



# ESCOLA NAVAL

*ta sainte & bief faire*



ASPOF M Martin Fedorchuk

## Otimização de trajetórias de veleiros

Estudo e desenvolvimento de algoritmos de otimização de trajetórias de veleiros

Dissertação para obtenção do Grau de Mestre em Ciências Militares Navais, na especialidade de Marinha



Alfeite  
2020





# ESCOLA NAVAL

talant de biefaire



ASPOF M Martin Fedorchuk

## Otimização de trajetórias de veleiros

Estudo e desenvolvimento de algoritmos de otimização de  
trajetórias de veleiros

Dissertação para obtenção do Grau de Mestre em  
Ciências Militares Navais, na especialidade de Marinha

Orientação de: CFR M Lourenço Gorricha

Co-orientação de: Prof. Miguel Ângelo Pereira Bento Moreira

*O Aluno Mestrando,*

\_\_\_\_\_  
ASPF M Martin Fedorchuk



*O Orientador,*

\_\_\_\_\_  
CFR M Lourenço Gorricha

*O Co-Orientador,*

\_\_\_\_\_  
Prof. Miguel Bento Moreira

Alfeite  
2020



“O importante não é o lugar que ocupamos, mas sim a direção na qual estamos a  
prosseguir.”

Lev Tolstói, 1896



Dedicado ao meu pai Vadym, à minha mãe Lesya e aos meus irmãos Uriil e Iegudiil.





# Agradecimentos

Agradeço profundamente a todos que, de alguma forma, contribuíram para os resultados da presente dissertação.

Agradeço:

- Aos meus orientadores, CFR M Lourenço Gorricha e Prof. Miguel Bento Moreira, pelo apoio, motivação, disponibilidade, proximidade e todos os conhecimentos transmitidos ao longo da elaboração da dissertação;
- Aos camaradas ASPOF Ribeiro Fraga e ASPOF Dias de Paiva com os quais passamos todo o período de confinamento na Escola Naval, pelo apoio e camaradagem nos tempo mais difíceis;
- Ao camarada ASPOF Afonso Ramos, pelo apoio, motivação e camaradagem demonstrados nos momentos necessários;
- Finalmente, mas não menos importante, aos meus pais, pela educação, apoio, carinho e dedicação ao longo de toda a minha vida.



# Resumo

## **Estudo e desenvolvimento de algoritmos de otimização de trajetórias de veleiros**

Desde o aparecimento de navios à vela, o objetivo principal destes era encontrar a melhor forma de aproveitamento do vento por forma a conseguir chegar com maior rapidez de um determinado ponto a outro. Sendo assim, o objetivo desta dissertação de mestrado é realizar um estudo e desenvolvimento de algoritmos de otimização de trajetórias para veleiros.

Inicialmente será realizado um estudo de diferentes abordagens e algoritmos já desenvolvidos nesta área, estudando também em que área estes podem ser utilizados. Seguindo-se um enquadramento teórico onde serão abordados assuntos como: a física na vela, a otimização e mais propriamente a técnica de simulação do recozimento (*Simulated Annealing*).

Com todos os conhecimentos reunidos será formulado o tipo de problema a resolver e explicada a criação do algoritmo de otimização para trajetórias com duas pernadas e vento constante, utilizando o programa *MATLAB*<sup>®</sup>, aplicando assim a técnica de simulação do recozimento.

Finalmente, será realizada a demonstração de algumas simulações numéricas, sendo abordados os resultados obtidos, analisando a sua veracidade e o funcionamento do algoritmo desenvolvido.

**Palavras-chave:** Otimização, Trajetória, Veleiro, Simulated Annealing, Matlab



# Abstract

## **Study and development of sailboat trajectory optimization algorithms**

Since the appearance of sailing ships, their main objective was to find the best way to harness the wind in order to be able to get more quickly from one point to another. Therefore, the objective of this master's dissertation is to carry out a study and development of trajectory optimization algorithms for sailboats.

Initially, a study of different approaches and algorithms already developed in this area will be carried out, also studying in which area they can be used. Following a theoretical framework where subjects such: as physics in sailing, optimization and more specifically the Simulated Annealing technique will be approached.

With all the knowledge gathered, the type of problem to be solved will be formulated and the creation of the optimization algorithm for trajectories with two kicks and constant wind will be explained, using the program *MATLAB*<sup>®</sup>, thus applying the Simulated Annealing technique.

Finally, the demonstration of some numerical simulations will be carried out, being approached the obtained results, analyzing their veracity and the performance of the developed algorithm.

**Keywords:** Otimization, Trajectory, Sailboat, Simulated Annealing, Matlab



# Prefácio

Este trabalho tem como objetivo o desenvolvimento dum algoritmo destinado a calcular a trajetória que minimize o tempo de trajeto dum veleiro, entre duas localizações geográficas.

Esta dissertação de mestrado encontra-se dividida em cinco capítulos fundamentais:

- No Capítulo 1 faremos uma introdução histórica e contextualizaremos o tema da otimização do tempo de trajeto à vela entre dois pontos;
- No Capítulo 2 apresentamos o enquadramento teórico dos diferentes tópicos que estão na base dos desenvolvimentos efetuados;
- No Capítulo 3 desenvolvemos, apresentamos e discutimos o algoritmo desenvolvido;
- No Capítulo 4 são apresentadas as simulações numéricas de teste do algoritmo desenvolvido utilizando diferentes cenários e simulações. Os resultados das simulações são discutidos com base nos conhecimentos de navegação à vela;
- Finalmente, no Capítulo 5 apresentamos as conclusões do trabalho elaborado e perspectivaremos possíveis desenvolvimentos para o trabalho futuro.





# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Motivação . . . . .	1
1.2	Vela Autônoma . . . . .	1
1.2.1	Vela robótica na Escola Naval . . . . .	2
1.3	Estado da Arte . . . . .	3
1.3.1	Início do Século XXI . . . . .	3
1.3.2	Algoritmo A* Utilizado pelo Veleiro Autônomo “ <i>Avalon</i> ” . .	5
1.3.3	Método de Otimização da Navegação a Curtas Distâncias Desenvolvida por Roland Stelzer . . . . .	7
1.3.4	Método de Isócronas Utilizado na Determinação da Melhor Trajetória . . . . .	8
1.4	Enquadramento do Problema . . . . .	9
<b>2</b>	<b>Enquadramento Teórico</b>	<b>11</b>
2.1	Distância, Tempo e Ângulo entre dois Pontos . . . . .	11
2.2	Trajetória na Vela . . . . .	12
2.3	Física da Vela . . . . .	12
2.3.1	Vento Aparente . . . . .	12
2.3.2	Forças Aplicadas num Veleiro em Movimento . . . . .	13
2.3.3	Orientação Ideal da Vela . . . . .	14
2.3.4	Parâmetros Geométricos de um Veleiro . . . . .	15
2.3.5	Velocidade do Veleiro em Função da Vela . . . . .	16
2.3.6	Diagrama Polar . . . . .	16
2.3.7	Conclusões . . . . .	17
2.4	Otimização . . . . .	18
2.4.1	Constituintes Básicos e Funcionamento de um Problema de Otimização . . . . .	18
2.4.2	Classificação de Algoritmos de Otimização de Acordo com o Método de Operação . . . . .	19
2.4.3	Técnicas e Algoritmos Heurísticos . . . . .	19
2.4.4	Simulação do Recozimento . . . . .	20
2.4.4.1	Minimização utilizando a Simulação de Recozimento . . . . .	21
<b>3</b>	<b>Formulação e Abordagem do Problema</b>	<b>23</b>
3.1	Formulação do Problema . . . . .	23
3.2	Abordagem do Problema (Instância A1) . . . . .	24
3.2.1	Parâmetros Iniciais . . . . .	24

3.2.2	Malha Base ( <i>grid</i> ) . . . . .	25
3.2.3	Cálculos Iniciais para duas Pernadas . . . . .	27
3.2.4	Geração do Diagrama Polar . . . . .	29
3.2.5	Cálculo da Velocidade e Tempo Necessário para Realização do Trajeto . . . . .	31
3.2.6	Preparação para a Utilização da Técnica de Simulação de Recozimento . . . . .	32
3.2.7	Início do ciclo da Simulação de Recozimento . . . . .	33
3.2.8	Demonstração de Resultados em Forma Gráfica . . . . .	34
3.2.9	Resumo da Abordagem do Problema . . . . .	35
<b>4</b>	<b>Simulações numéricas</b>	<b>39</b>
4.1	Geração do Gráfico de Evolução do Tempo Mínimo Obtido com o Decorrer das Épocas . . . . .	39
4.2	Geração do Melhor Trajeto para Diferentes Veleiros, Usando o Mesmo Cenário . . . . .	40
4.3	Geração do Melhor Trajeto para um Veleiro, Usando Cenários Diferentes . . . . .	44
4.3.1	Navegação à Bolina Cerrada . . . . .	44
4.3.2	Navegação com o Vento pelo Través . . . . .	45
4.3.3	Navegação ao Largo . . . . .	47
4.3.4	Navegação com o Vento pela Popa . . . . .	49
4.4	Conclusões preliminares . . . . .	51
<b>5</b>	<b>Conclusões e Trabalho Futuro</b>	<b>53</b>
5.1	Conclusões Gerais . . . . .	53
5.2	Trabalhos Futuros . . . . .	54
	<b>Bibliografia</b>	<b>55</b>
	<b>Apêndices</b>	<b>59</b>
<b>A</b>	<b>Terminologia Náutica</b>	<b>59</b>
<b>B</b>	<b>Diagramas polares de velocidade</b>	<b>61</b>
B.1	Diagrama polar do veleiro “ <i>Albin 79</i> ” . . . . .	61
B.1.1	Forma Gráfica . . . . .	61
B.1.2	Forma programada ( <i>gera_diag_polar_albin</i> ) . . . . .	62
B.2	Diagrama polar do N.R.P. “Sagres” . . . . .	63
B.2.1	Forma gráfica . . . . .	63
B.2.2	Forma programada ( <i>gera_diag_polar_sagres</i> ) . . . . .	64
B.3	Diagrama polar do veleiro <i>Alba</i> . . . . .	65
B.3.1	Forma programada ( <i>gera_diag_polar_alba</i> ) . . . . .	65
B.4	Diagrama polar dos veleiro tipo <i>SB 20</i> . . . . .	66
B.4.1	Forma programada ( <i>gera_diag_polar_sb20</i> ) . . . . .	66
<b>C</b>	<b>Programa principal e subprogramas</b>	<b>67</b>
C.1	Programa principal ( <i>main_mf.m</i> ) . . . . .	67

C.2	Subprogramas . . . . .	74
C.2.1	Malha base ( <i>malhabase.m</i> ) . . . . .	74
C.2.2	Rumo <i>A-B</i> ( <i>theta0P.m</i> ) . . . . .	74
C.2.3	Trajeto <i>A-P-B</i> ( <i>trajeto1P.m</i> ) . . . . .	75
C.2.4	Rumo <i>A-P</i> e <i>P-B</i> ( <i>theta1P.m</i> ) . . . . .	76
C.2.5	Geração do diagrama polar ( <i>gera_diag_polar</i> ) . . . . .	76
<b>D</b>	<b>Gráficos</b>	<b>77</b>
D.1	Navegação contra o vento do veleiro do tipo “Albin 79” . . . . .	77
D.2	Navegação contra o vento do veleiro do tipo “SB 20” . . . . .	78
D.3	Navegação contra o vento do N.R.P. “Sagres” . . . . .	79
D.4	Navegação à bolina cerrada do N.R.P. “Sagres” . . . . .	80
D.5	Navegação com o vento pelo través do veleiro N.R.P. “Sagres” . . . . .	81
D.6	Navegação com o vento na alheta do veleiro N.R.P. “Sagres” . . . . .	82
D.7	Navegação à popa do veleiro N.R.P. “Sagres” . . . . .	83
	<b>Anexos</b>	<b>85</b>
<b>I</b>	<b>Ficha técnica do veleiro “SB 20”</b>	<b>85</b>
<b>II</b>	<b>Tabela de velocidades do veleiro “Alba”</b>	<b>89</b>



# Lista de Figuras

1.1	Veleiro autónomo “FAST” desenvolvido pela FEUP. (Cruz e Alves (2014)) . . . . .	2
1.2	Veleiro <i>Barlavento</i> depois das alterações em 2019. (Santos (2019)) .	3
1.3	Visualização do funcionamento do programa dinâmico desenvolvido por Philpott e Manson (Philpott e Manson (2001)) . . . . .	4
1.4	Veleiro autónomo “Avalon” desenvolvido pelo Instituto Federal da Suíça (Erckens et al. (2010)). . . . .	5
1.5	Exemplos de planeamentos de navegação a favor do vento (a) e contra o vento (b) com obstáculos utilizados no veleiro “Avalon” (Erckens et al. (2010)). . . . .	7
1.6	Determinação do rumo ótimo numa trajetória contra o vento usando o método de Stelzer (Stelzer (2012)). . . . .	8
1.7	Simulação da determinação do melhor trajeto utilizando o método de isócronas utilizando o <i>software qtVlm</i> (Rabaud (2016)). . . . .	9
2.1	Relação entre as velocidades real ( $\vec{v}$ ) e aparente ( $\vec{v}_{ap}$ ) do vento e a velocidade do veleiro ( $\vec{V}$ ) (Barros e Sá, 2005) . . . . .	13
2.2	Forças aplicadas sobre um veleiro pelo vento e pelo mar (Barros e Sá, 2005) . . . . .	13
2.3	Orientação ideal da vela consoante o vento aparente e a proa do veleiro. (imagem adaptada de Barros e Sá, 2005) . . . . .	15
2.4	Diagrama polar para diferentes valores de $\sigma$ . (imagem adaptada de Barros e Sá, 2005) . . . . .	16
2.5	Diagrama polar do veleiro da marca <i>Albin</i> (imagem adaptada de <a href="http://www.sailonline.org">www.sailonline.org</a> ) . . . . .	17
2.6	Trajetória em bordos na navegação contra o vento (imagem adaptada de Barros e Sá, 2005) . . . . .	17
2.7	Zona de não navegabilidade (imagem adaptada de Alves, 2016) . . .	18
2.8	Diferentes tipos de algoritmos de otimização (Weise, 2009) . . . . .	20
2.9	Exemplo de um gráfico de evolução dos resultados consoante o número de iterações usando o método de Simulação de Recozimento (imagem adaptada de <a href="https://www.github.com/chncyhn/simulated-annealing-tsp">www.github.com/chncyhn/simulated-annealing-tsp</a> ). .	21
2.10	Fluxograma da Simulação de Recozimento. . . . .	22
3.1	Criação dos parâmetros iniciais. . . . .	25
3.2	Exemplo do funcionamento da <i>grid</i> como espaço de procura. . . .	26
3.3	Programação da malha base ( <i>grid</i> ). . . . .	26

3.4	Transformação dos ângulos do rumo de $A$ para $B$ e do ângulo do vento para coordenadas polares. . . . .	27
3.5	Programação para obtenção das distâncias e rumos do ponto $A$ para $P$ e de $P$ para $B$ . . . . .	28
3.6	Transformação dos ângulos do rumo de $A$ para $P$ e de $P$ para $B$ e do ângulo do vento para coordenadas polares. . . . .	28
3.7	Diagrama polar em forma gráfica do N.R.P. "Sagres" (imagem adaptada de da Mota, 2014). . . . .	29
3.8	Programação do diagrama polar do N.R.P. "Sagres". . . . .	30
3.9	Cálculo das velocidades e tempo necessário entre os ponto $A - P - B$ . . . . .	31
3.10	Organização dos trajetos e das coordenadas correspondentes por ordem crescente de tempo. . . . .	32
3.11	Guardar os melhores $4 \times \sqrt{\text{linhas} \times \text{colunas}}$ e o melhor resultado na altura. . . . .	32
3.12	Guarda do melhor resultado da geração anterior e diminuição do espaço de perturbação. . . . .	33
3.13	Criação dos vetores perturbantes e a realização da perturbação. . . . .	33
3.14	Realização de novos cálculos após a perturbação das posições. . . . .	34
3.15	Fase final de cada geração e a criação da variável para geração de um gráfico ( <i>tempo_min</i> ). . . . .	34
3.16	Criação do gráfico de evolução do menor tempo com o decorrer das épocas. . . . .	35
3.17	Criação do gráfico com os trajeto $A - B$ e a trajetória ótima $A - P - B$ . . . . .	35
3.18	Fluxograma do <i>script</i> realizado. . . . .	36
3.19	Pseudocódigo do <i>script</i> realizado. . . . .	37
4.1	Evolução do tempo mínimo obtido com o decorrer de 1000 ciclos. . . . .	39
4.2	Evolução do tempo mínimo obtido com o decorrer de 100 ciclos. . . . .	40
4.3	Navegação contra o vento do veleiro "Albin 79", com uma velocidade do vento de 13 nós. . . . .	41
4.4	Navegação contra o vento do veleiro "SB 20", com uma velocidade do vento de 13 nós. . . . .	41
4.5	Navegação contra o vento do N.R.P. "Sagres", com uma velocidade do vento de 13 nós. . . . .	42
4.6	Janela de comando do <i>MATLAB</i> ® com os resultados obtidos para o veleiro "SB 20". . . . .	43
4.7	Trajetória ótima de navegação do N.R.P. "Sagres" à bolina cerrada. . . . .	45
4.8	Trajetória ótima de navegação do N.R.P. "Sagres" com o vento pelo través. . . . .	46
4.9	Trajetória ótima de navegação do N.R.P. "Sagres" com vento a entrar por um ângulo de $115^{\circ}$ . . . . .	48
4.10	Trajetória ótima de navegação do N.R.P. "Sagres" com vento a entrar por um ângulo de $140^{\circ}$ . . . . .	48
4.11	Trajetória ótima de navegação do N.R.P. "Sagres" com o vento pela popa. . . . .	50

# Lista de Tabelas

4.1	Parâmetros iniciais para a navegação contra o vento. . . . .	40
4.2	Distâncias a percorrer, tempo necessário e velocidades dos veleiros na navegação contra o vento, com uma velocidade do vento de 13 nós. . . . .	42
4.3	Parâmetros iniciais para a navegação à bolina cerrada. . . . .	44
4.4	Distâncias, tempos e velocidades da navegação do N.R.P. "Sagres" à bolina cerrada com uma e duas pernadas. . . . .	44
4.5	Parâmetros iniciais para a navegação com o vento pelo través. . . . .	46
4.6	Distâncias, tempos e velocidades da navegação do N.R.P. "Sagres" com vento pelo través. . . . .	46
4.7	Parâmetros iniciais para a navegação ao largo. . . . .	47
4.8	Distâncias, tempos e velocidades da navegação do N.R.P. "Sagres" com vento a entrar por um ângulo de $115^{\circ}$ . . . . .	47
4.9	Distâncias, tempos e velocidades da navegação do N.R.P. "Sagres" com vento a entrar por um ângulo de $140^{\circ}$ . . . . .	47
4.10	Parâmetros iniciais para a navegação com o vento pela popa. . . . .	49
4.11	Distâncias, tempos e velocidades da navegação do N.R.P. "Sagres" com vento pela popa. . . . .	50





# Lista de Abreviaturas, Acrónimos e Símbolos

## Abreviaturas

MATLAB® MATrix LABoratory

## Acrónimos

*PP* Polinómio por partes

*SR* Simulação do Recozimento

2D Duas Dimensões

3D Três Dimensões

ASC Autonomous Surface Craft

EN Escola Naval

FEUP Faculdade de Engenharia da Universidade do Porto

GPS Global Positioning System

IA Inteligência Artificial

N.R.P. Navio da República Portuguesa

USV Unmanned Surface Vehicles

## Símbolos

$\alpha$  Ângulo entre o vetor da velocidade do vento e a vela [°]

$\beta$  Orientação da vela [°]

$\beta_{ideal}$  Orientação ideal da vela [°]

$\eta$  Constante de diferença entre a proa e o rumo do veleiro

$\gamma$	Abatimento do veleiro	$[^{\circ}]$
$\phi$	Ângulo	$[^{\circ}]$
$\rho_{agua}$	Densidade da água	$[kg/m^3]$
$\rho_{ar}$	Densidade do ar	$[kg/m^3]$
$\sigma$	Constante de velocidade do veleiro consoante a área vélica e a área de casco imerso	
$\theta$	Rumo do veleiro	$[^{\circ}]$
$\theta_{ap}$	Rumo aparente do veleiro	$[^{\circ}]$
$\Delta t$	Variação do tempo	$[s]$
$\vec{v}_{ap}$	Velocidade aparente do vento	$[m/s]$
$\vec{V}$	Velocidade do veleiro	$[m/s]$
$\vec{v}$	Velocidade do vento	$[m/s]$
$A$	Posição inicial	
$A_L$	Área longitudinal do casco	$[m^2]$
$a_L$	Área longitudinal da vela	$[m^2]$
$A_T$	Área transversal do casco do veleiro	$[m^2]$
$a_T$	Área transversal da vela	$[m^2]$
$B$	Posição final	
$C$	Constante	
$E$	Épocas	
$f$	Função objetivo	
$F_L$	Força longitudinal exercida pela água no veleiro	$[N]$
$f_L$	Força longitudinal exercida pelo vento no veleiro	$[N]$
$F_T$	Força transversal exercida pela água no veleiro	$[N]$
$f_T$	Força transversal exercida pelo vento no veleiro	$[N]$
$P$	Ponto intermédio onde é realizada a guinada.	

$R_T$	Raio da Terra	$[km]$
$s$	Distância	$[m]$
$t$	Tempo	$[s]$
$v$	Velocidade	$[m/s]$
$X$	Domínio da função	
$x$	Variável	



# Capítulo 1

## Introdução

### 1.1 Motivação

Esta dissertação tem como objetivo desenvolver um algoritmo de otimização do tempo de trajeto em navegação à vela entre duas posições geográficas.

O interesse na otimização de rotas de veleiros surge no contexto da utilização, cada vez mais frequente, de veículos autônomos para atividades onde estes podem trazer vantagens acrescidas, como por exemplo a recolha de dados meteorológicos, oceanográficos, hidrográficos ou acústicos com vista a melhorar a extensão do nosso conhecimento relacionado com o mar em toda a sua profundidade, sendo que a utilização destes veículos aparece igualmente no contexto da área da segurança e defesa, onde também está a registar uma grande atenção.

Desde que, historicamente, se iniciou o aproveitamento do vento para conseguir chegar, com maior rapidez, de um determinado ponto ao outro, e até aos dias de hoje, a navegação à vela tem registado importantes desenvolvimentos técnicos que permitiram não só velejar mais rapidamente como também utilizar menores ângulos de bolina, sendo que a compreensão científica mais profunda e completa da física da vela tem permitido também melhorar a execução técnica desta atividade.

Mais recentemente, com o aparecimento de veleiros autônomos, e com o desenvolvimento da vela robótica, as capacidades e experiência do velejador deixaram de ter relevância, tornando-se necessário substituir esses elementos por capacidades autônomas que permitam, de forma sistemática e automatizada escolher as melhores soluções para definir, a cada momento, o melhor caminho. Este problema conduz-nos ao principal objetivo desta dissertação de mestrado, que é desenvolver, testar e discutir um algoritmo com o objetivo de minimizar o tempo de percurso por parte dum veleiro autónomo na realização dum trajeto de um ponto inicial para um final.

### 1.2 Vela Autónoma

A vela autónoma ou robótica pode ser definida como todo o processo de navegação que é realizado autonomamente pelo veleiro usando os seus sistemas (Stelzer,

2012). A utilização do vento como fonte de propulsão dos veleiros não é elementar. A otimização do tempo de trajeto entre dois pontos, assume uma particular importância na vela autónoma.

As características chave da vela robótica foram resumidas por *Stelzer, 2012* como:

- O vento é a única fonte de propulsão;
- Não é remotamente controlado, ou seja, todo o sistema de controlo encontra-se a bordo;
- É completamente auto-suficiente energeticamente, sendo esta parte não obrigatória, mas que abre um maior alcance de oportunidades de desenvolvimento.

O desenvolvimento de veículos de superfície não tripulados, em inglês *Unmanned Surface Vehicles* (USV), teve o seu início no ano de 1993 com um dos primeiros *Autonomous Surface Craft* (ASC) “ARTEMIS”, desenvolvido na *MIT Sea Grant College Program* (Manley, 2008). Ao longo dos anos, diferentes tipos de USVs foram criados com diferentes objetivos e para realização de diferentes atividades. De assinalar o desenvolvimento por parte da Faculdade de Engenharia da Universidade do Porto (FEUP) de um prototipo denominado “FASt” (Fig.1.1), que consiste “num pequeno veleiro de 2,5 metros, totalmente autónomo que consegue navegar autonomamente entre uma sequência de *waypoints*” (Cruz & Alves, 2014).



FIGURA 1.1: Veleiro autónomo “FASt” desenvolvido pela FEUP.  
(Cruz e Alves (2014))

### 1.2.1 Vela robótica na Escola Naval

O desenvolvimento de dissertações de mestrado por parte da Escola Naval (EN) no âmbito da vela robótica teve o seu início no ano de 2010 com a criação do projeto “*eVentos*” (Cavaco (2011)), coordenado pelo Professor Doutor Victor Lobo (Lobo (2014)).

A base das dissertações iniciais era um veleiro *Laser RC* que foi reconvertido para navegação autónoma. Após vários testes foi identificado que este veleiro

não possuía robustez nem espaço para todo o *payload* desenvolvido. Assim em 2015 foi desenvolvido um veleiro de raiz capaz de responder às necessidades pretendidas, ficando esse veleiro com o nome de “Barlavento” (Santos (2019)). Em 2019 foi desenvolvida mais uma dissertação de mestrado onde houve melhorias no veleiro construído em 2015, tanto a nível estrutural como a nível de *software* (Santos (2019)).



FIGURA 1.2: Veleiro *Barlavento* depois das alterações em 2019. (Santos (2019))

De referir, no âmbito do projeto eVentos, a realização de várias parcerias com a FEUP por parte da EN no desenvolvimento do *FAST* referido anteriormente (Fernandes (2016)).

## 1.3 Estado da Arte

Neste subcapítulo vão ser abordados diferentes métodos de otimização e escolha de rotas para veleiros desenvolvidos por diferentes autores, tanto no âmbito da vela autónoma como relacionados com a vela tradicional. Iremos dar mais ênfase a rotas de pequena distância, visto que é o tipo de rotas mais semelhante ao tipo de trajetórias que vão ser abordadas nesta dissertação de mestrado.

### 1.3.1 Início do Século XXI

Em 2001 foi realizado um estudo de otimização de rotas de veleiros por parte de Andy Philpott e Andrew Mason que propuseram dois modelos diferentes de cálculo com o objetivo de minimizar o tempo necessário para chegar do ponto de partida ao ponto de chegada, sendo um modelo para trajetórias de curto alcance e outro para trajetórias de longo alcance. Neste caso vai se proceder apenas à análise do método de cálculo da trajetória de pequeno alcance, pela razão já descrita na introdução à este subcapítulo.

O modelo usa um método de programação dinâmica, em que o problema é dividido em várias etapas e a decisão adotada na resolução de cada etapa depende da decisão tomada na parte anterior. O efeito de todas as decisões tomadas são englobados num conjunto de variáveis, utilizadas na determinação da decisão ótima

numa dada parte (Philpott e Manson (2001)). Para a determinação das diferentes velocidades do veleiro para as diferentes condições ambientais é utilizado o diagrama polar desse veleiro, uma explicação das funcionalidades de um diagrama polar pode encontrar-se no Sub-subcapítulo 2.3.6.

Os parâmetros iniciais que dão origem à resolução do problema são:

- Coordenadas do ponto inicial e final;
- A hora de início;
- A direção e velocidade do vento e da corrente iniciais.

Para a divisão do espaço e distribuição da trajetória em pernadas é criado um algoritmo que gera uma matriz dividida em nós, sendo que cada vez que é percorrida uma distância de um nó a outro as condições ambientais são recalculadas nesse nó, a velocidade do vento é transformada numa nova variável que é a revertida da força média constante do vento e a direção do vento é alterada através da aplicação de cadeias de Markov<sup>1</sup> em que os estados dessas cadeias são um conjunto de ângulos discretos (Philpott e Manson (2001)).

O tempo mínimo calculado para navegar até ao destino é calculado a partir de cada nó em que passa a trajetória na matriz, sendo escolhido como próximo nó para se dirigir aquele que apresente o menor tempo para se chegar ao nó de chegada e assim sucessivamente. Na Figura 1.3 está explicado como este programa dinâmico funcionava, ficando o caminho com menor custo é representado com uma seta mais grossa.

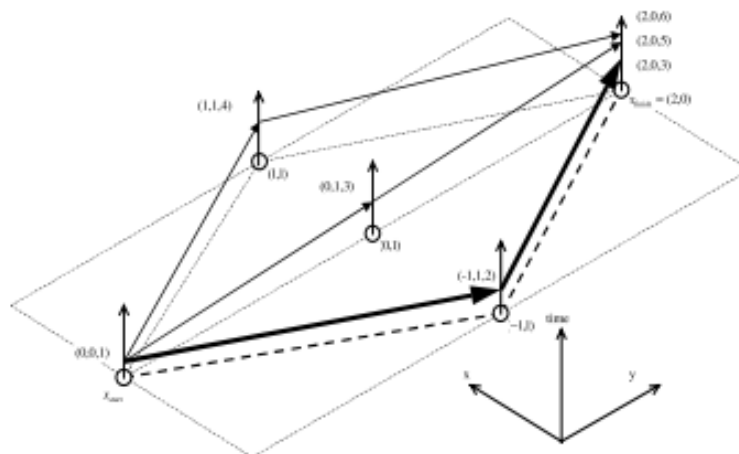


FIGURA 1.3: Visualização do funcionamento do programa dinâmico desenvolvido por Philpott e Manson (Philpott e Manson (2001))

<sup>1</sup>Cadeia de Markov: é um processo estocástico com estados discretos com a propriedade de que a distribuição de probabilidade do próximo estado depende apenas do estado atual e não na sequência de eventos que o precederam (Meyen e Tweedie (2005)).



Na Figura 1.3 verificamos também que a ilustração está feita num domínio tridimensional (3D) em que  $x$  e  $y$  são as coordenadas do ponto e a coordenada  $z$  corresponde ao tempo necessário para chegar ao ponto.

#### 1.3.2 Algoritmo A\* Utilizado pelo Veleiro Autônomo “Avalon”

O veleiro autônomo “Avalon” foi desenvolvido pelo Instituto Federal da Tecnologia na Suíça com o objetivo de percorrer o Oceano Atlântico autonomamente sem qualquer influência humana (Giger et al. (2009)).

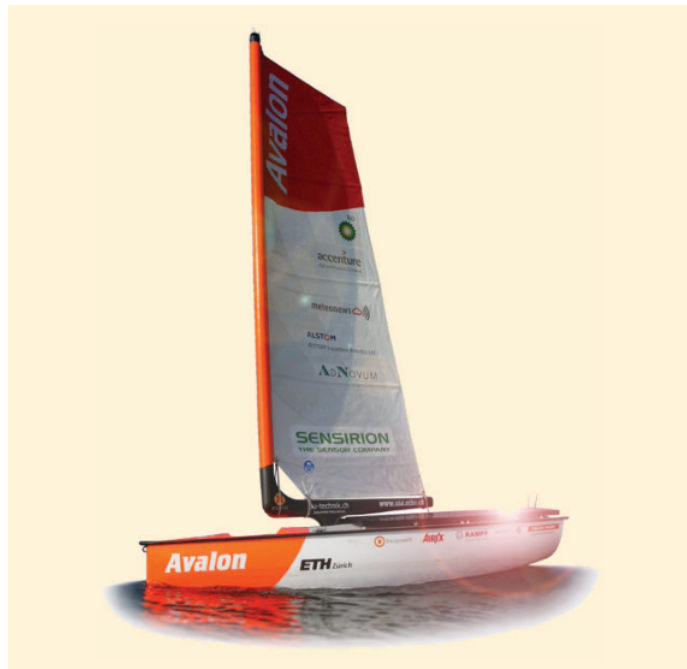


FIGURA 1.4: Veleiro autônomo “Avalon” desenvolvido pelo Instituto Federal da Suíça (Erckens et al. (2010)).

Quanto ao algoritmo de planeamento da rota a adotar pelo veleiro esta divide-se em duas partes: um planeamento global e um planeamento local. O planeamento global é realizado fixando *waypoints* baseados no conhecimento, clima e análises de correntes (Erckens et al. (2010)). No que toca ao planeamento local, este passa a ser usado para ir da posição até ao próximo *waypoint* do planeamento global, usando os dados obtidos a partir dos sensores do veleiro.

O planeamento local consiste num algoritmo A\*<sup>2</sup> em que ao começar no ponto inicial da rede o algoritmo percorre todo o espaço até chegar ao último ponto dessa rede, sendo que a cada ponto percorrido é atribuído um custo. A trajetória ótima é então conseguida ao ser sempre seguido o ponto vizinho com o menor custo a começar no ponto final até ao ponto inicial da rede (Erckens et al. (2010)).

---

<sup>2</sup>Algoritmo A\*: é um algoritmo de inteligência artificial (IA) usado na pesquisa heurística, em que a pesquisa é feita dividindo o espaço de procura numa rede em que o centro de cada quadrado da rede é um nó (Peng, Huang e Luo (2015)).

Para fazer a deslocação de um ponto da rede para o vizinho é utilizado um plano em 3D em que as coordenadas  $x$  e  $y$  são as coordenadas geográficas e a coordenada  $z$  corresponde ao rumo do veleiro na posição em que se encontra. A forma como é realizada a transformação das coordenadas geográficas para cartesianas é demonstrada pelas equações 1.1e 1.2, sendo que esta transformação assume que a superfície à linha de água é plana, visto que este tipo de planeamento é feito para pequenas distâncias.

$$x = R_T \times \cos(latitude) \times \frac{\pi}{180} \times longitude \quad (1.1)$$

$$y = R_T \times \frac{\pi}{180} \times latitude \quad (1.2)$$

Em que  $R_T$  é o raio da Terra e a latitude e longitude obtidos do GPS (*Global Positioning System*) em graus. Para criar a rede local o mais pequena possível, o algoritmo posiciona a origem o mais próximo possível da coordenada inicial (Erckens et al. (2010)).

O custo geral dado a cada movimento é, portanto, o tempo de navegação para o próximo nó, calculado ao dividir a distância até ao próximo nó pela velocidade do veleiro. Para obter resultados razoáveis foram implementados adicionalmente os seguintes custos:

- tempo de navegação desde o nó inicial;
- tempo estimado de navegação até ao nó final (heurística);
- custo da mudança de rumo;
- custo da manobra (virar por d'avante<sup>3</sup> ou virar em roda<sup>4</sup>);
- custo de compensação a partir da conexão direta entre o nó inicial e o final.

A obtenção da velocidade do veleiro é feita novamente, tal como no método apresentado anteriormente, através do diagrama polar do veleiro.

De referir que este método tem também desenvolvida uma componente que para além do cálculo da trajetória em relação às condições climáticas altera a rota se houver algum obstáculo no seu seguimento, como podemos observar na Figura 1.5, onde são demonstrados dois diferentes tipos de planeamento de acordo com as condições do vento e tendo em atenção os obstáculos ao longo da rota.

<sup>3</sup>Virar por d'avante: Forma de mudar o bordo pelo qual entra o vento no veleiro, orçando, fazendo passar o vento pela proa do veleiro (Schmid (1978)).

<sup>4</sup>Virar em roda: Forma de mudar o bordo pelo qual entra o vento no veleiro, arribando, fazendo passar o vento pela popa do veleiro (Schmid (1978)).

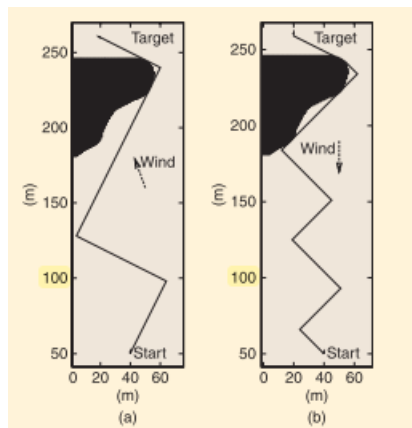


FIGURA 1.5: Exemplos de planeamentos de navegação a favor do vento (a) e contra o vento (b) com obstáculos utilizados no veleiro “Avalon” (Erckens et al. (2010)).

### 1.3.3 Método de Otimização da Navegação a Curtas Distâncias Desenvolvida por Roland Stelzer

Este método, visto que foi desenvolvido quase na mesma altura que o método descrito no Sub-subcapítulo 1.3.2, apresenta semelhanças na fase inicial da sua implementação com aquele que foi desenvolvido para o “Avalon”. A forma como é realizada transformação das coordenadas e a utilização do diagrama polar do veleiro para estudo do comportamento do veleiro são algumas dessas semelhanças.

Como é lógico, na realização de uma navegação de um ponto inicial até ao a um ponto final é necessário considerar a direção (rumo) na qual é necessário navegar e o vento (direção e velocidade). O objetivo deste método é diminuir a distância ao ponto final o mais rápido possível (Stelzer (2012)). Numa abordagem analítica o método faz com que o veleiro navegue a um rumo com uma velocidade ótima local durante um período de tempo até atingir um ponto onde um rumo alternativo proporciona uma velocidade ótima global. O veleiro guina e navega à velocidade ótima global até atingir um rumo que novamente prove ser significativamente melhor, havendo assim uma nova guinada, e assim sucessivamente. Como podemos perceber neste método é considerado que o vento é constante durante todo o trajeto.

Na Figura 1.6 podemos verificar que (a) representa uma situação de navegação onde o ponto  $T$  é o ponto final e o vento provém do mesmo setor para o qual é necessário navegar. O propósito do algoritmo é identificar o rumo que apresente a melhor velocidade para o veleiro, quando é encontrado um rumo com velocidade maior é esse o rumo que passa a ser seguido. Na situação (b) da mesma figura (1.6) verificamos as mesmas condições que em (a), mas numa fase mais tardia do trajeto. Nesta situação é atingida a uma parte em que há dois rumos que apresentem a mesma velocidade máxima conseguida, isto acontece quando o vento provém diretamente da direção do ponto final  $T$ . Neste caso em que há várias alternativas é aplicada uma condição de histerese (Stelzer (2012)).

Resumindo, o objetivo do algoritmo é calcular o rumo ao qual é conseguida

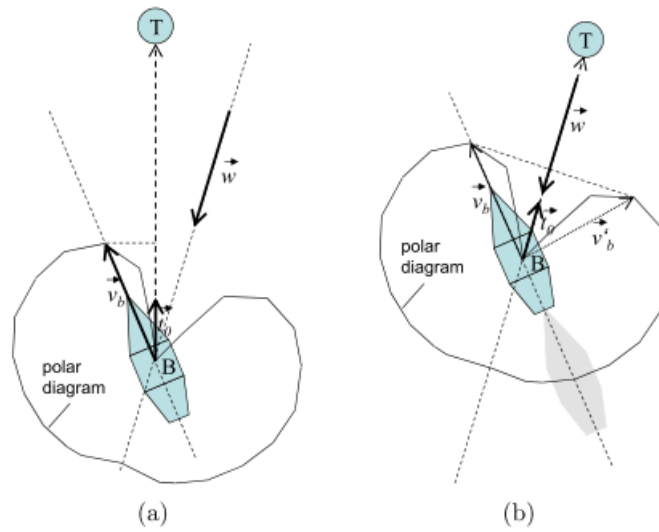


FIGURA 1.6: Determinação do rumo ótimo numa trajetória contra o vento usando o método de Stelzer (Stelzer (2012)).

a maior velocidade na direção do ponto de chegada, em forma de ângulo (que pode ser transformado em vetor). O algoritmo é chamado constantemente em etapas de tempo razoáveis, com o objetivo de atualizar o rumo proposto à medida que os parâmetros mudem (Stelzer (2012)).

### 1.3.4 Método de Isócronas Utilizado na Determinação da Melhor Trajetória

Neste método, o conhecimento das características do diagrama polar do veleiro é o primeiro passo para determinar a trajetória ótima. É feita também uma simplificação que considera que as condições meteorológicas são perfeitamente conhecidas, ou seja, sabe-se a direção e a intensidade do vento  $W(l, t)$  em qualquer localização  $l$  e instante  $t$  (Rabaud (2016)).

O método de isócronas é baseado na determinação, a qualquer momento, da curva das maiores distâncias alcançáveis em qualquer direção a partir da posição inicial do veleiro. O conhecimento do vetor do vento e a informação recolhida do diagrama polar do veleiro fornecem todas as posições alcançáveis para qualquer direção,  $\theta$  após um pequeno intervalo de tempo,  $\Delta t$ , permitindo obter  $l = V(\theta)\Delta t$ . Assim sendo, obtém-se todas as posições alcançáveis para todas as direções ( $\theta$ ) que no seu conjunto definem a primeira curva isócrona para um determinado  $\Delta t$ . De notar que o  $\Delta t$  não deve ser muito grande devido às variações espaciais das condições meteorológicas. Após a primeira iteração o processo é repetido em cada posição  $l$  ao longo da isócrona previamente determinada, o diagrama de velocidade polar do veleiro correspondente à intensidade do vento  $W(l, t + \Delta t)$  é desenhado e orientado consoante a direção do vento. O processo é assim repetido até ao instante em que a ultima isócrona cruza com o ponto final (Rabaud (2016)). O trajeto com o tempo mínimo de execução é então determinado começando no ponto final e unindo todos

os pontos em cada isócrona correspondentes à velocidade máxima à que o veleiro consegue navegar no sentido pretendido.

Um exemplo da aplicação deste método pode ser observado na Figura 1.7, em que é realizada uma simulação utilizando o programa *qtVlm*<sup>5</sup>. Nesta simulação o trajeto dum veleiro que necessita de realizar uma navegação entre uma posição de partida oceânica e a cidade de Nova Iorque. As cores de fundo correspondem às condições meteorológicas inicialmente previstas, as curvas são as linhas isócronas e a melhor trajetória é a linha que une os diferentes pontos das isócronas, de acordo com o critério referido anteriormente.

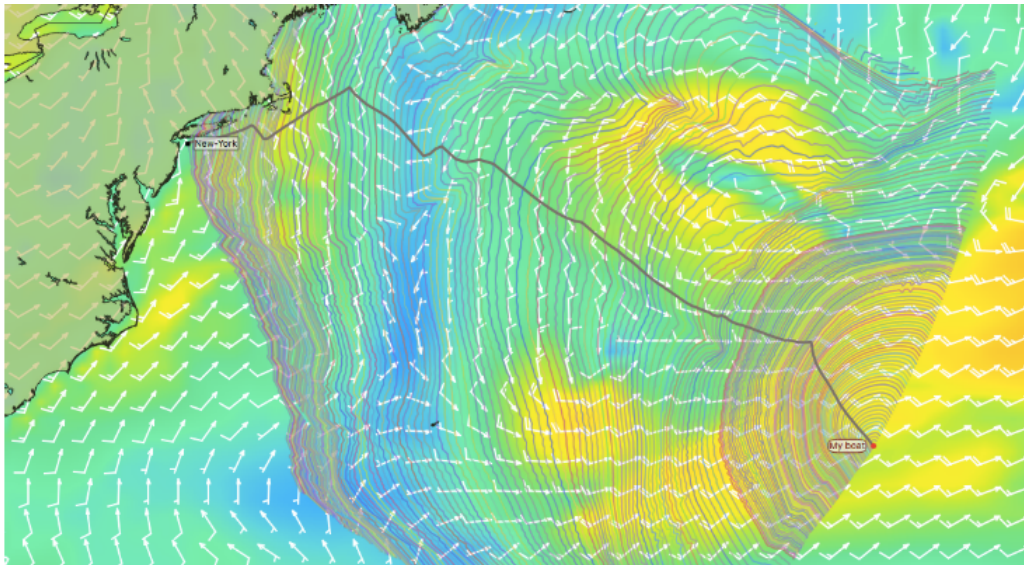


FIGURA 1.7: Simulação da determinação do melhor trajeto utilizando o método de isócronas utilizando o *software qtVlm*(Rabaud (2016)).

## 1.4 Enquadramento do Problema

O problema abordado nesta dissertação de mestrado consiste em encontrar, para qualquer tipo de veleiro, a melhor trajetória a seguir com o objetivo de navegar a partir de um ponto inicial  $A$ , para um ponto final  $B$ . Em relação ao número de pernadas apenas vão ser estudados trajetos com duas pernadas e o vento vai ser considerado invariável durante todo o trajeto, não sendo considerados os efeitos da corrente sobre o veleiro.

Em relação à metodologia de resolução vai-se proceder ao uso de uma técnica de otimização denominada de Simulação de Recozimento, descrita no Subcapítulo 2.4.4. A sua aplicação e no desenvolvimento de todos os algoritmos usaremos o *MATLAB*<sup>6</sup> que consiste num *software* com uma linguagem de alto desempenho

---

<sup>5</sup>*qtVlm*: É um *software* projetado para veleiros que opera com vários tipos de previsão de condições meteorológicas (Manual of *qtVlm* version 5.9.10).

<sup>6</sup>*MATLAB*®: *MATrix LABoratory*

para computação técnica, que engloba a computação, a visualização e a programação (MathWorks (2005)).

# Capítulo 2

## Enquadramento Teórico

O objetivo deste capítulo é estabelecer as ferramentas necessárias para o desenvolvimento do trabalho. Abordaremos em particular os conceitos de: cálculo de distâncias e ângulos em duas dimensões (2D), a física relacionada com a vela e, por fim, uma breve explicação sobre a otimização e os seus diferentes aspetos. Apresentaremos também a técnica de Simulação de Recozimento utilizada do desenvolvimento deste trabalho.

### 2.1 Distância, Tempo e Ângulo entre dois Pontos

Para se calcular o tempo mais curto no trajeto de um ponto para outro, é primeiro necessário conhecer a distância ( $s$ ) entre esses dois pontos. Essa distância, num sistema cartesiano, pode ser calculada a partir da fórmula 2.1 onde  $x_A$  e  $y_A$  são coordenadas cartesianas de um determinado ponto inicial  $A$  e  $x_B$  e  $y_B$  são as coordenadas de um ponto final  $B$ .

$$s_{AB} = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2} \quad (2.1)$$

No seguimento, para se calcular o tempo ( $t$ ) necessário para se deslocar entre o ponto inicial e o ponto final utiliza-se a seguinte expressão:

$$t_{AB} = \frac{s_{AB}}{v} \quad (2.2)$$

Onde  $v$  é a velocidade a que se move um determinado objeto entre os pontos inicial e final e  $s_{AB}$  representa a distância retilínea percorrida .

Como sabemos em qualquer arte de navegação é necessário saber o rumo a tomar para se dirigir de um ponto para outro, daí a importância de saber que o ângulo definido entre esses pontos relativamente a um referencial fornece-nos o dito rumo. Para se conhecer o ângulo ( $\phi$ ) entre dois pontos , num sistema cartesiano, é necessário recorrer à seguinte expressão:

$$\phi_{AB} = \arctan\left(\frac{y_B - y_A}{x_B - x_A}\right) \times \frac{180}{\pi} \quad (2.3)$$

Em que os pontos  $x_A$ ,  $y_A$ ,  $x_B$  e  $y_B$  são os mesmos que os descritos acima.

## 2.2 Trajetória na Vela

Na vela a escolha da trajetória é um aspeto muito importante visto que é essa escolha que decide o tempo a realizar um determinado trajeto. Na navegação à vela a trajetória mais curta raramente é a mais rápida, dependendo da maré e das condições meteorológicas. Assim sendo, é necessário sempre procurar minimizar o tempo a percorrer de um ponto inicial  $A$  até ao ponto final  $B$ , como é traduzido na equação 2.4, em vez de procurar minimizar a distância entre esses dois pontos (Rabaud, 2016).

$$T_{AB} = \int_A^B dt = \int_A^B \frac{ds}{v} \quad (2.4)$$

Em que  $T_{AB}$  corresponde ao tempo total a percorrer do ponto  $A$  até ao ponto  $B$ .

## 2.3 Física da Vela

Todo este subcapítulo é baseado no trabalho de Barros e Sá, 2005 que aborda vários aspetos da física da navegação à vela. Para uma perceção mais completa de todos os cálculos e conclusões presentes neste subcapítulo recomenda-se a recorrer ao artigo mencionado acima.

### 2.3.1 Vento Aparente

A força motriz dum veleiro é o vento. Esta ação é equilibrada pelas forças de resistência do mar e a aceleração produzida. Em resultado da aceleração, a velocidade aumenta até que a força motriz do vento é completamente equilibrada pela resistência. Nessa fase a velocidade,  $\vec{V}$  do veleiro passa a ser constante. Podemos definir o rumo do veleiro como  $\theta$ , sendo este o ângulo entre a velocidade do vento,  $\vec{v}$  e o simétrico da velocidade do veleiro,  $-\vec{V}$  (Barros e Sá, 2005), como podemos observar na Figura 2.1.

Por outro lado  $\vec{v}$  e  $\vec{v}_{ap}$  relacionam-se pela seguinte expressão:

$$\vec{v}_{ap} = \vec{v} - \vec{V}. \quad (2.5)$$



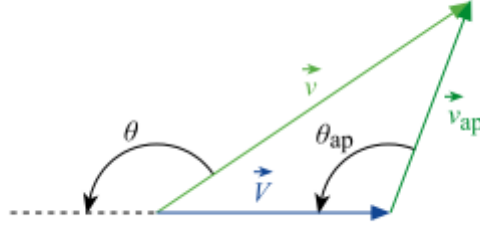


FIGURA 2.1: Relação entre as velocidades real ( $\vec{v}$ ) e aparente ( $\vec{v}_{ap}$ ) do vento e a velocidade do veleiro ( $\vec{V}$ ) (Barros e Sá, 2005)

Com base na Figura 2.1 podemos verificar também uma relação entre  $\theta$  e  $\theta_{ap}$  que pode ser traduzida por:

$$\cotan\theta_{ap} = \cotan\theta + \frac{V}{v\sin\theta} \quad (2.6)$$

$$v_{ap}^2 = v^2 + 2vV\cos\theta + V^2. \quad (2.7)$$

### 2.3.2 Forças Aplicadas num Veleiro em Movimento

A partir da Figura 2.2 podemos observar a representação das forças horizontais que atuam num veleiro, utilizando um sistema de eixos cartesianos em que o eixo  $xx$  é a direção da vela.

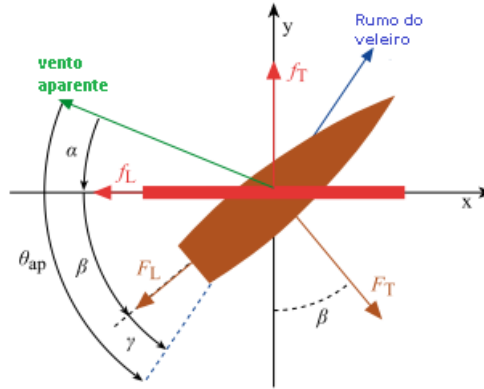


FIGURA 2.2: Forças aplicadas sobre um veleiro pelo vento e pelo mar (Barros e Sá, 2005)

Podemos também verificar na Figura 2.2 que  $f_T$  e  $f_L$  são as forças exercidas pelo vento no veleiro na transversal e longitudinal e,  $F_T$  e  $F_L$  forças análogas, mas exercidas pelo mar. A nível de ângulos,  $\alpha$  é o ângulo contado a partir do vetor da velocidade do vento até à vela,  $\beta$  é aquele que é contado entre a vela e a linha

de popa do veleiro e  $\gamma$  é o ângulo contado da linha de popa até ao simétrico do vetor da velocidade do veleiro, podendo este ângulo também ser denominado como abatimento<sup>7</sup> do veleiro. Conhecendo estes ângulos podemos concluir que a soma de todos os ângulos anteriormente mencionados fornece-nos o rumo aparente do veleiro,  $\theta_{ap}$ , tal como podemos verificar na equação 2.8 .

$$\theta_{ap} = \alpha + \beta + \gamma \quad (2.8)$$

Segundo a Figura 2.2 podemos concluir que as componentes da força total exercida sobre um veleiro são definidas por:

$$F_x = F_T \sin \beta - F_L \cos \beta - f_L = \rho_{agua} A_L V^2 (\sin \gamma \sin \beta - \cos \gamma \cos \beta) - \rho_{ar} a_L v_{ap}^2 \cos \alpha \quad (2.9)$$

$$F_y = f_T - F_T \cos \beta - F_L \sin \beta = \rho_{ar} a_L v_{ap}^2 \sin \alpha - \rho_{agua} A_L V^2 (\sin \gamma \cos \beta + \cos \gamma \sin \beta) \quad (2.10)$$

Em que  $a_L$  é a área longitudinal da vela,  $A_L$  é a área longitudinal do casco,  $\rho_{agua}$  e  $\rho_{ar}$  são as densidades da água e o ar correspondentemente.

Quanto ao abatimento do veleiro,  $\gamma$ , pode ser calculado a partir da seguinte equação:

$$\tan \gamma = \frac{\rho_{agua} A_T V^2 \cos \beta + \rho_{ar} a_T v_{ap}^2 \cos(\theta_{ap} - \beta)}{\rho_{agua} A_T V^2 \sin \beta - \rho_{ar} a_T v_{ap}^2 \sin(\theta_{ap} - \beta)}. \quad (2.11)$$

Em que  $A_T$  é a área transversal do casco do veleiro e  $a_T$  é a área transversal da vela.

### 2.3.3 Orientação Ideal da Vela

A orientação ideal é aquela que fornece ao veleiro a velocidade máxima dadas as circunstâncias da velocidade do vento  $v$  e do rumo  $\theta$ , que se mantêm constantes ( $dv = d\theta = 0$ ) (Barros e Sá, 2005). Após algumas simplificações algébricas chega-se ao resultado que estabelece que a orientação ideal é quando esta é colocada precisamente na bissetriz do ângulo entre as direções do vento aparente e do simétrico da proa do veleiro (Barros e Sá, 2005)(Figura 2.3) , ou seja:

$$\beta_{ideal} = \frac{\theta_{ap}}{2}. \quad (2.12)$$

Onde  $\beta_{ideal}$  é a orientação ideal da vela.

---

<sup>7</sup>Abatimento: é o ângulo que mede o desvio do rumo do veleiro em relação à direção da sua quilha. (Barros e Sá, 2005)

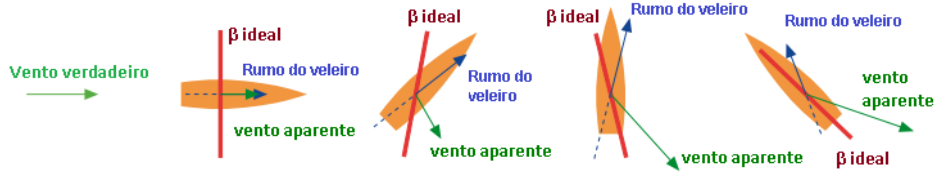


FIGURA 2.3: Orientação ideal da vela consoante o vento aparente e a proa do veleiro. (imagem adaptada de Barros e Sá, 2005)

### 2.3.4 Parâmetros Geométricos de um Veleiro

Um bom desempenho do veleiro é resultado da forma longilínea da vela e do casco, ou seja,  $a_L \gg a_T$  e  $A_L \gg A_T$  (Barros e Sá, 2005). Apesar do casco de um veleiro não ser tão esbelto como uma vela, a relação entre estas áreas longitudinal e transversal é ampliada pelo patilhão<sup>8</sup> (Barros e Sá, 2005).

A área vélica dum veleiro é superior à área transversal imersa do casco mas a densidade do ar é bastante inferior à da água, assim temos:

$$\rho_{ar}a_T \ll \rho_{ar}A_T \approx \rho_{agua}A_T \ll \rho_{agua}A_L. \quad (2.13)$$

A equação 2.11 pode assumir a forma

$$\gamma = \arctan\left(\eta \cotan \frac{\theta_{ap}}{2}\right), \quad (2.14)$$

obtendo-se a expressão que fornece a velocidade do veleiro em função da velocidade do vento, do rumo e da orientação da vela:

$$V^2 = \sigma^2 v_{ap}^2 \sin^2 \frac{\theta_{ap}}{2}. \quad (2.15)$$

Os parâmetros geométricos do veleiro,  $\eta$  e  $\sigma$  são definidos pelas seguintes expressões:

$$\eta = \frac{A_T}{A_L} \quad (2.16)$$

e

$$\sigma = \sqrt{\frac{\rho_{ar}a_L}{\rho_{agua}A_T}}. \quad (2.17)$$

A constante  $\eta$  é diminuta pois  $a_L \ll A_T$ . A constante  $\sigma$  é da ordem da unidade ou inferior e indica o quão veloz é o veleiro (aumenta com a área da vela e diminui com a área imersa do casco do veleiro)(Barros e Sá, 2005).

<sup>8</sup>Patilhão: Uma extensão da quilha com uma área transversal mínima mas com uma área longitudinal substancial. (Barros e Sá, 2005)

### 2.3.5 Velocidade do Veleiro em Função da Vela

Como já foi explicado,  $\theta_{ap}$  e  $v_{ap}$  são funções de  $V$ ,  $v$  e  $\theta$ . Ao introduzirmos os resultados obtidos das equações 2.6 e 2.7 na equação 2.15 podemos resolvê-la por forma a obter a velocidade do veleiro,  $V$  em função da velocidade do vento,  $v$  e do rumo do veleiro,  $\theta$  (Barros e Sá, 2005), a equação resultante terá de ser elevada ao quadrado para a raiz quadrada ser eliminada, obtendo-se:

$$V = g(\theta)\sigma v. \quad (2.18)$$

Podemos observar que  $V$  aumenta linearmente com  $v$ , sendo a dependência do rumo descrita pela solução da equação polinomial quadrática,  $g(\theta)$ , que é descrita por:

$$4(1 - \sigma^2)g^4 - 8\sigma\cos\theta g^3 + (\sigma^2\sin^2\theta - 4)g^2 + 2\sigma\cos\theta\sin^2\theta g + \sin^2\theta = 0. \quad (2.19)$$

### 2.3.6 Diagrama Polar

Um diagrama polar de um veleiro é uma forma conveniente de representar os dados de previsão de velocidade, sendo extremamente útil pois ajuda-nos a perceber a relação dos três fatores mais importantes no desempenho de um veleiro: a velocidade do vento, o ângulo do vento e a velocidade do veleiro (*OFFSHORE RACING CONGRESS (ORC) - Speed Guide Explanation*, 2009). Como os veleiros geralmente são simétricos grande parte dos diagramas polares apresentam a informação relativa a um sector com amplitude de 180 graus (Rabaud, 2016), como se pode observar na Figura 2.5. Na Figura 2.4 podemos verificar um diagrama polar gerado a partir das equações 2.18 e 2.19 que nos fornece a velocidade do veleiro em função do rumo e diferentes valores da constante  $\sigma$ .

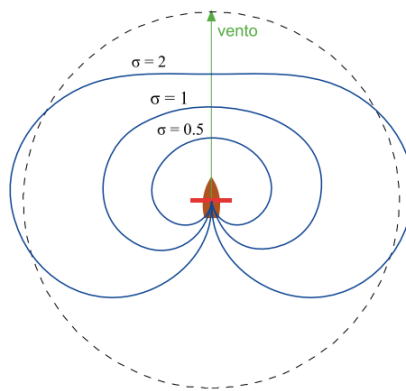


FIGURA 2.4: Diagrama polar para diferentes valores de  $\sigma$ . (imagem adaptada de Barros e Sá, 2005)

Os diagramas polares podem obter-se experimentalmente em provas de mar durante as quais a leitura das velocidades é feita diretamente para diferentes velocidades e ângulos de vento e os valores são registados no diagrama como podemos ver na Figura 2.5 que corresponde ao diagrama polar do veleiro do tipo “Albin 79”.

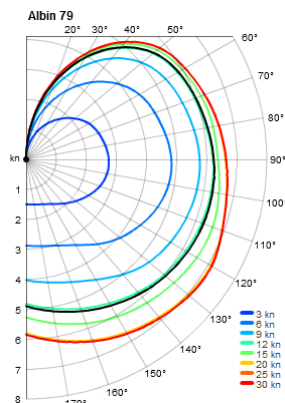


FIGURA 2.5: Diagrama polar do veleiro da marca *Albin* (imagem adaptada de [www.sailonline.org](http://www.sailonline.org))

### 2.3.7 Conclusões

O autor, no seu artigo, estabelece algumas conclusões respeitantes ao rumo a tomar para diferentes ângulos e velocidades do vento. Quando  $\sigma$  é elevado a maior velocidade que o veleiro pode atingir não se encontra no rumo a favor do vento, mas sim algures entre esse rumo e a sua perpendicular, como se pode observar na Figura 2.4. Quando os valores de  $\sigma$  são diminutos, ou seja quando  $\sigma \rightarrow 0$ , podemos observar na Figura 2.4 que a velocidade máxima é atingida no rumo a favor do vento.

No que toca na navegação contra o vento chega-se à conclusão que para garantir uma maior rapidez no trajeto de um ponto para outro é necessário realizar bordos<sup>9</sup> como podemos verificar na Figura 2.6 onde se navega com uma diferença de cerca de  $45^\circ$  em relação ao vento, ou aquele que proporciona a maior velocidade na direção aposta ao vento fazendo mudanças de rumo de  $90^\circ$ .

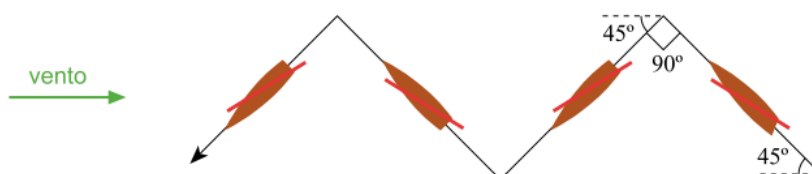


FIGURA 2.6: Trajetória em bordos na navegação contra o vento (imagem adaptada de Barros e Sá, 2005)

<sup>9</sup>Bordos: Trajetória realizada em ziguezague alternando o bordo pelo qual entra o vento no veleiro.

Pode-se afirmar que existe sempre um setor no qual é impossível a realização da navegação, sendo essa zona formada por um arco entre os 40 e os 45º para cada lado do rumo contrário ao do vento, dependendo sempre das características do veleiro, como se pode observar na Figura 2.7.

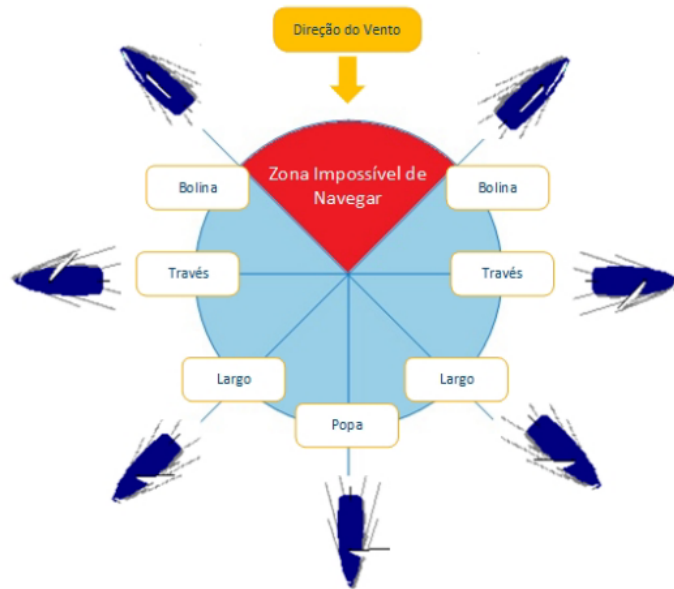


FIGURA 2.7: Zona de não navegabilidade (imagem adaptada de Alves, 2016)

## 2.4 Otimização

“Um dos princípios fundamentais da vida é a procura de uma solução ideal e, enquanto existe a humanidade, estaremos sempre à procura da perfeição nas diferentes áreas, querendo alcançar o máximo grau de felicidade usando o mínimo de esforço possível” (Weise, 2009), a este processo de procura podemos chamar de “Otimização”. Apesar de esta definição parecer fácil ela esconde muitos problemas complexos, por exemplo, uma solução pode consistir em diferentes tipos de dados, restrições não-lineares podem restringir o espaço de pesquisa, o mesmo espaço de pesquisa pode apresentar várias soluções. As características do problema podem mudar com o tempo, sendo estes apenas alguns problemas que ajudam a explicar algumas das complexidades de um processo de otimização (Engelbrecht, 2007). Na sua generalidade, podemos afirmar que os problemas de otimização dividem-se em dois tipos: os que procuram o máximo e os que procuram o mínimo.

### 2.4.1 Constituintes Básicos e Funcionamento de um Problema de Otimização

Matematicamente, um problema de otimização engloba os seguintes constituintes:

- Uma função objetivo  $f : X \rightarrow Y$ , em que o contradomínio  $Y$  tem de ser um subconjunto de números reais ( $Y \subseteq \mathbb{R}$ ) e o domínio  $X$  da função é chamado o espaço de pesquisa (Weise, 2009);
- Um conjunto incógnitas ou variáveis, que afetam o valor da função objetivo. Se  $x$  representar as incógnitas, também conhecidas como variáveis independentes, então  $f(x)$  quantifica a qualidade da solução candidata (Engelbrecht, 2007);
- Um conjunto de constantes, que limitam os valores que podem ser atribuídos às incógnitas (Engelbrecht, 2007).

Resumindo, o objetivo da otimização é descobrir no espaço de procura  $X$  o máximo ou o mínimo da função objetivo  $f$  cumprindo com todas as restrições impostas pelo conjunto de constantes Engelbrecht, 2007. Para realizar a procura desses máximos ou mínimos são utilizados os denominados algoritmos de otimização.

### 2.4.2 Classificação de Algoritmos de Otimização de Acordo com o Método de Operação

Os algoritmos de otimização, como podemos ver na Figura 2.8, podem classificar-se em determinísticos e os probabilísticos. Nos algoritmos determinísticos o “caminho” para determinar o ótimo é único (Weise, 2009). Quanto aos algoritmos probabilísticos estes são utilizados quando o tipo de problema não permite utilizar uma resolução determinística. Como o próprio nome diz só garantem um resultado em probabilidade, ou seja, é garantido que o resultado ótimo global vai ser encontrado, mas o tempo de computação até esse resultado ser obtido pode ir até infinito (Liberti e Kucherenko, 2005).

Na Figura 2.8 podemos observar uma possível classificação dos algoritmos probabilísticos.

### 2.4.3 Técnicas e Algoritmos Heurísticos

Nos algoritmos probabilísticos matemáticos podem ser utilizadas técnicas heurísticas. As técnicas heurísticas podem ser definidas como aquelas que são usadas para simular matematicamente os problemas do dia a dia e resolvê-las através de um método de “tentativa e erro”, não garantindo totalmente que o resultado ótimo vai ser encontrado (Mijwil, 2015). Apesar de não estar totalmente garantida totalmente a obtenção do resultado ótimo, Martí e Reinelt, 2011 destaca no seu livro algumas razões e necessidades para o uso destas técnicas:

- A necessidade de encontrar boas soluções para problemas difíceis;
- Quando não é conhecido nenhum método determinístico para a resolução do problema, ou quando este método é conhecido, mas não pode ser implementado no *software* ou *hardware disponível*;
- Uma técnica heurística é mais flexível, possibilitando fazer uma incorporação de condições difíceis de modelar.

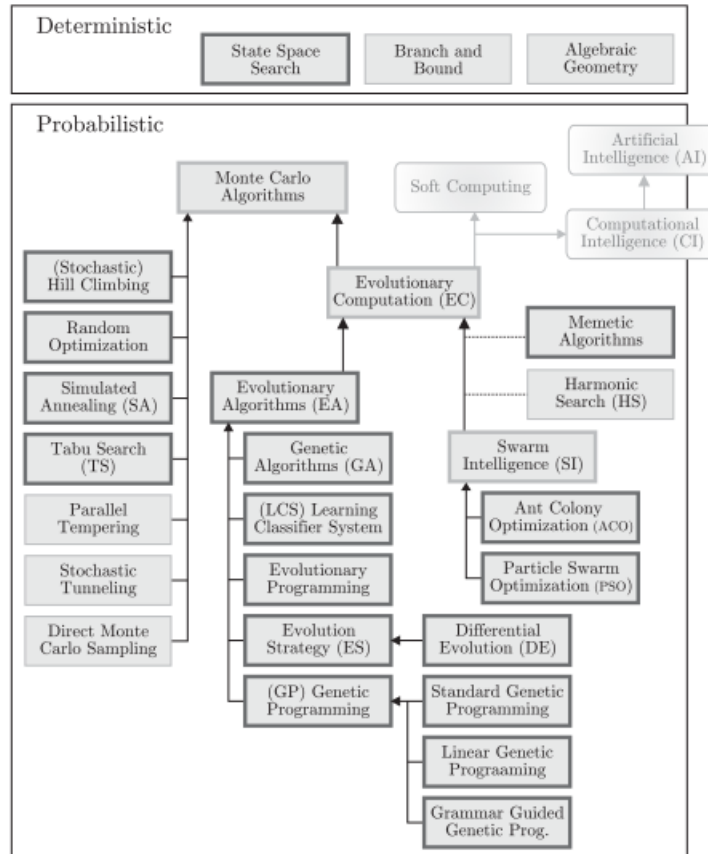


FIGURA 2.8: Diferentes tipos de algoritmos de otimização (Weise, 2009)

Os requisitos acabados de apresentar justificam a utilização dum algoritmo heurístico que vai ser utilizado no desenvolvimento desta dissertação de mestrado.

Os algoritmos heurísticos são normalmente programados para multi-uso, ou seja, de modo a existir um método adequado para uso em diferentes problemas de otimização (Husmann, Lange e Spiegel, 1999). Neste momento existem diferentes tipos de algoritmos heurísticos, destacando-se:

- Algoritmos Genéticos (*Genetic Algorithms*);
- Inteligência de Enxame (*Swarm Intelligence*);
- Pesquisa *Tabu* (*Tabu Search*);
- Simulação do Recozimento (*Simulated Annealing*);

Nesta dissertação utilizaremos a técnica de Simulação de Recozimento.

#### 2.4.4 Simulação do Recozimento

A Simulação do Recozimento (SR), como já foi referido, é uma técnica heurística de otimização que foi proposta por Kirkpatrick, Gelatt e Vecchi em 1983 (Johnson, Aragon e Mcgeoch, 1989). O método de SR é uma técnica utilizada



para problemas de otimização em larga escala, especialmente nos quais o extremo (máximo ou mínimo) local encontra-se escondido entre muitos extremos locais mais “pobres” (Press et al., 1992).

### 2.4.4.1 Minimização utilizando a Simulação de Recozimento

Visto que o objetivo da dissertação de mestrado é realizar uma minimização do tempo, vai ser abordado o uso do SR numa minimização, da mesma forma como este vai ser utilizado na resolução do problema da dissertação.

Inicialmente, vai ser definido o problema de otimização. Como foi explicado no Subcapítulo 2.4.1, num problema de otimização a função que se pretende otimizar é habitualmente designada por “função objetivo”,  $f$  (Kirkpatrick, Gelatt e Vecchi, 1983). O domínio dessa função,  $X$  normalmente é denominado por “espaço de procura” onde se encontram todas as variáveis,  $x$  que vão ser utilizadas na otimização de  $f$ . Pode também existir um conjunto de constantes que, de alguma forma, podem restringir o valor dado às incógnitas, no caso deste problema este conjunto de constantes não é aplicável. É também definida uma variável  $E$  que corresponde quantidade de iterações (épocas) que vai ser aplicada a SR ao longo do processo.

A forma de funcionamento e a aplicação da técnica para a realização da minimização estão representados no fluxograma da Figura 2.10.

Na Figura 2.9 é demonstrada uma evolução do resultado mínimo obtido aplicando a SR ao longo das iterações ou épocas,  $E$ .

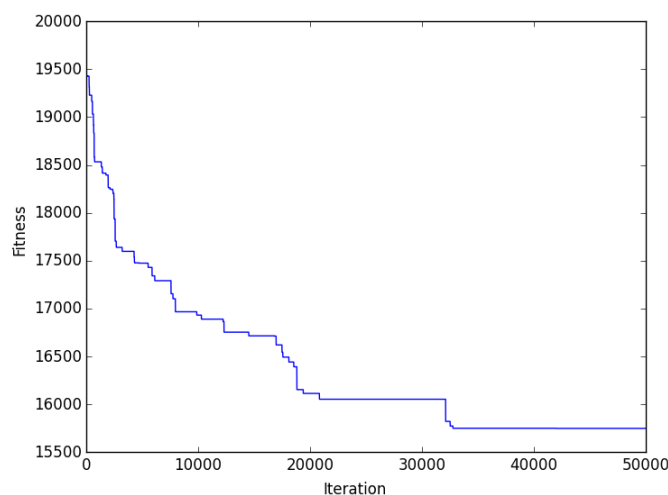


FIGURA 2.9: Exemplo de um gráfico de evolução dos resultados consoante o número de iterações usando o método de Simulação de Recozimento (imagem adaptada de [www.github.com/chncyhn/simulated-annealing-tsp](https://www.github.com/chncyhn/simulated-annealing-tsp)).

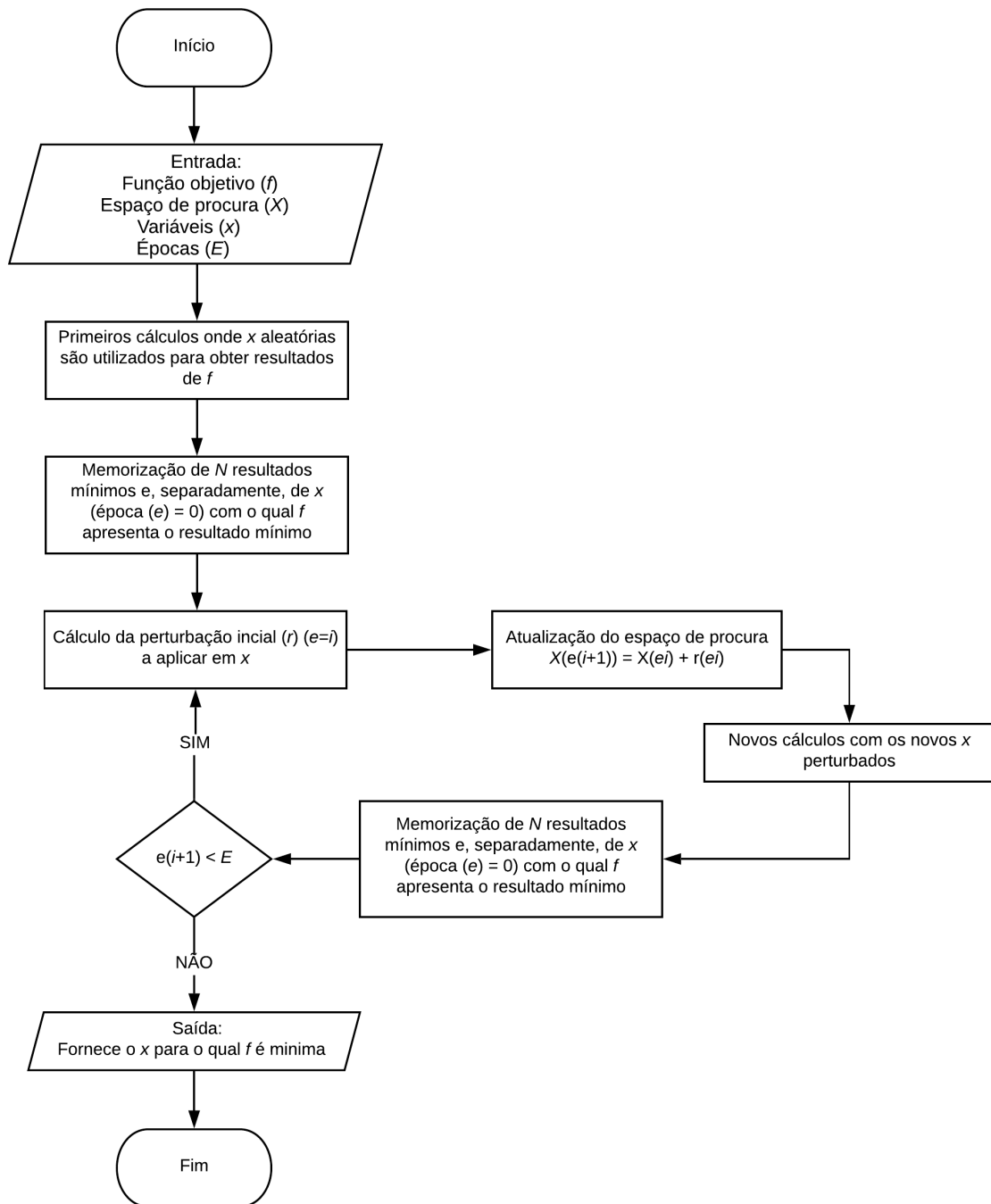


FIGURA 2.10: Fluxograma da Simulação de Recozimento.

# Capítulo 3

## Formulação e Abordagem do Problema

Neste capítulo serão formulados os vários níveis de dificuldade, seguindo-se da caracterização do problema cuja resolução será abordada através de um *script*<sup>10</sup> utilizando o *software MATLAB*®.

### 3.1 Formulação do Problema

O problema a resolver nesta dissertação consiste em calcular a trajetória ótima para a navegação de um veleiro desde um ponto inicial  $A$  até um ponto final  $B$ .

Como se trata de navegação à vela é necessário ter sempre em consideração algumas variáveis não só relacionadas com as condições meteorológicas, mas também com fenómenos locais. Como sabemos o vento é a principal força que movimenta o veleiro, daí a direção e a força do vento serem os principais fatores influenciadores considerados neste tipo de problemas. Também é necessário ter em consideração o estado do mar e a força da corrente que no caso deste problema ficam como fatores de segundo plano, finalmente, mas não menos importante é a limitação geográfica, visto que uma trajetória que a nível matemático é a melhor pode passar por locais onde a navegação é impossível.

O problema a resolver pode ser caracterizado recorrendo a formulações com diferentes níveis de profundidade:

- **Instância A1** - Esta instância é a que apresenta um nível básico de complexidade. Aqui consideramos a direção e a força do vento constante ao longo de todo o trajeto,  $V = V(t) = C$ ;
- **Instância A2** - Esta instância pode ser considerada uma continuação da primeira visto o vento é estacionário, mas pode variar geograficamente;
- **Instância B** - Aqui já é apresentado um nível mais avançado de complexidade onde já deve ser considerado uma direção e força do vento independente da

---

<sup>10</sup>Script: Conjunto de instruções em código ("script", in Dicionário Priberam da Língua Portuguesa [em linha], 2008-2020, <https://dicionario.priberam.org/script>).

posição, sendo estes não estacionários durante o trajeto de  $A$  para  $B$  com um conhecimento determinista antecipado,  $V = V(t)$ ;

- **Instância C** - Nesta instância o nível de complexidade é novamente aumentado e a direção e a velocidade do vento passam a ser dependentes da posição e não estacionários, mas continuam a ser de conhecimento determinista antecipado,  $V = V(x, y, t)$ , onde  $x$  e  $y$  corresponde às coordenadas da posição do veleiro;
- **Instância D** - Finalmente, esta instância apresenta o maior nível de complexidade, visto que engloba todos os parâmetros da instância anterior, mas é só conhecido o panorama, a nível de condição do vento, presente, sendo este incerto no futuro.

Nesta dissertação abordaremos a instância A1 não considerando os efeitos da corrente, do estado de mar nem entrando em conta com as perdas de velocidade nas guinadas.

## 3.2 Abordagem do Problema (Instância A1)

Neste subcapítulo vai ser explicada detalhadamente a forma como foi abordado o problema através do programa *MATLAB*<sup>®</sup>. De notar que o programa está dividido num programa principal (*main\_mf*) e num conjunto de sub-programas que são utilizados do decorrer do programa principal. Na explicação da metodologia utilizada depois da explicação e criação de diferentes variáveis utilizadas nos cálculos segue entre parênteses o nome atribuído a essa variável no *script* para uma melhor percepção do leitor. De referir que o programa completo e os subprogramas podem ser consultados no *Apêndice C*.

### 3.2.1 Parâmetros Iniciais

Iniciando o *script* é necessário definir os parâmetros iniciais, ou de entrada, sendo esses parâmetros:

- As coordenadas do ponto  $A$  e  $B$ , de notar que vai ser utilizado um sistema de coordenadas cartesianas, visto que nesta fase inicial não é necessária a utilização de coordenadas geográficas, sendo que, se futuramente for necessário, realizar a transformação de coordenadas geográficas para cartesianas pode ser feito através das fórmulas apresentadas no subcapítulo 1.3.2;
- A direção (*apov*) e velocidade (*vento*) do vento;
- Uma variável denominada “passos” que definirá uma distância de referência (distância entre linhas e colunas) na geração da malha base (*passos*).

A forma como foi criado o início deste *script* pode ser observado na Figura 3.1.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% main_mf
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc; close all; clear all;
%
iprint=0; % FAZER iprint=1 para gravar os gráficos produzidos
%
x1=0; y1=0; % Coordenadas do Ponto A
x2=100; y2=0; % Coordenadas do ponto B
%
A=[x1 y1]; % PONTO A
B=[x2 y2]; % PONTO B
%
%Velocidade do vento em nós
vento=13;
% Imposição do valor do Azimute Para Onde Sopra o Vento ("aposv") que
% permitirá definir o ângulo theta correspondente ao rumo de cada pernada,
% (rAP e rPB) para entrada no diagrama polar das velocidades do veleiro.
aposv=270; %Azimute para onde sopra o vento
A=[x1 y1]; % PONTO A
B=[x2 y2]; % PONTO B
%
% A variável "passos" definirá uma distância de referência "resolucao"
% para gerar a malha base (distância entre "linhas" e "colunas") e para
% a "perturbação" de representantes da poluplação de trajectos possíveis.
% A distância de referência = dAB/passos;
passos=5; % número de subdivisões de dAB (distância de referência)

```

FIGURA 3.1: Criação dos parâmetros iniciais.

#### 3.2.2 Malha Base (*grid*)

Após serem inseridos os parâmetros iniciais é necessário proceder à criação de uma malha base (*malhabase*) que vai representar o nosso espaço de procura. Esta *grid*<sup>11</sup> vai ser dividida em duas matrizes: uma com as coordenadas  $x$  dos pontos ( $Xbase$ ) e outra com as coordenadas  $y$  ( $Ybase$ ). A determinação do espaço de procura segue uma lógica em que: é criado um vetor onde à coordenada  $x$  mínima entre os pontos  $A$  e  $B$  ( $menorX$ ) é subtraída a distância, em milhas náuticas entre esses pontos ( $dAB$ ) e à coordenada  $x$  máxima ( $maiorX$ ) essa distância é somada, seguindo a mesma lógica para as coordenadas máxima e mínima  $y$ . A distância entre cada ponto (*resolucao*) é calculada utilizando a variável *passos*, em que esta distância corresponde à distância entre os pontos  $A$  e  $B$  é dividida pela variável “passos”. Após os vetores ( $Xvector$  e  $Yvector$ ) serem criados é utilizada a função *meshgrid*<sup>12</sup> que cria uma *grid* como podemos observar na Figura 3.2. A forma como esta parte foi programada segue na Figura 3.3.

<sup>11</sup>Grid: Sistema de rede em quadrículas.

<sup>12</sup>Meshgrid: Função que transforma o domínio especificado por um único vetor ou dois vetores  $x$  e  $y$  em matrizes  $X$  e  $Y$  para uso na avaliação de funções de duas variáveis. As linhas de  $X$  são cópias do vetor  $x$  e as colunas de  $Y$  são cópias do vetor  $y$  (MathWorks, 2005).

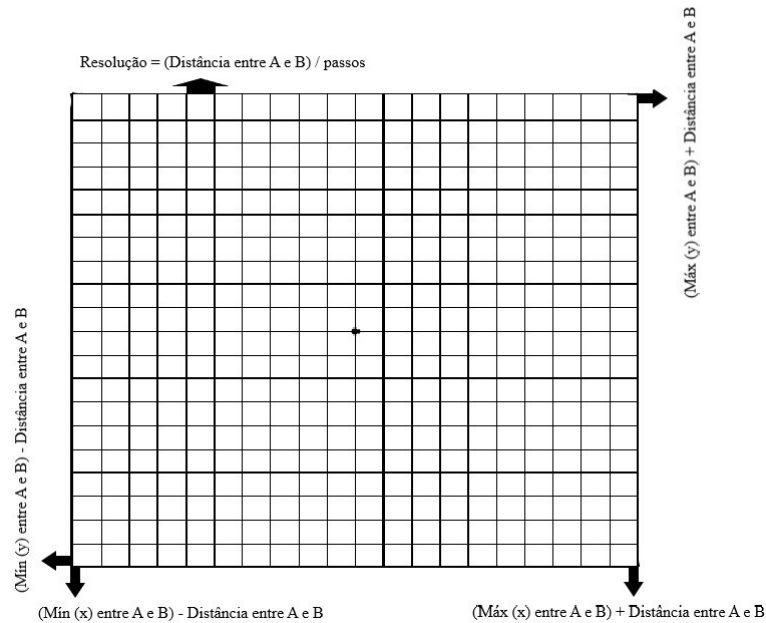


FIGURA 3.2: Exemplo do funcionamento da *grid* como espaço de procura.

```
function [Xbase, Ybase, dAB, rAB, resolucao]=malhabase(A,B,passos)
% dAB-distância de A a B
% rAB-rumo de A a B (coordenadas polares [-180 a +180])
dAB=pdist2(A,B,'euclidean');
rAB=atan2d(B(2)-A(2),B(1)-A(1));
%Resolução que devine a distância entre os pontos na malha
resolucao=dAB/passos;
%Eixos da malha
menorX=min(A(1),B(1));
maiorX=max(A(1),B(1));
menorY=min(A(2),B(2));
maiorY=max(A(2),B(2));
%Criação da malha
vectorX=menorX-dAB:resolucao:maiorX+dAB;
vectorY=menorY-dAB:resolucao:maiorY+dAB;
%
[Xbase,Ybase] = meshgrid(vectorX,vectorY);
end
```

FIGURA 3.3: Programação da malha base (*grid*).

Nesta parte também é calculado o rumo a tomar para realizar uma navegação direta do ponto  $A$  para  $B$  ( $rAB$ ) que vai ser utilizado nos cálculos mais a diante, de notar que este rumo vai ser convertido em coordenadas polares para mais tarde servir como variável de entrada no diagrama polar mudando o seu nome de variável ( $thetaAB$ ). A Figura 3.4 apresenta o processo da conversão dos ângulos dos rumos entre  $A$  e  $B$ , mostrando também a conversão do ângulo de onde sopra o vento ( $apovs$ ) ficando este como ângulo de onde sopra o vento ( $adosv$ ), igualmente para coordenadas polares.

```
function [thetaAB, adosv]=thetaOP(rAB,aposv)
%rAB rumo de A para B em coordenadas polares [-180 a +180]
%aposv-Azimuth Para Onde Sopra o Vento
%adosv-Azimuth De Onde Sopra o Vento
% thetaAB Ângulos de entrada no diagrama polar

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Conversão de "aposv" (0[N] a 360) em "adosv" [-180 0[E] +180[
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

aposv_cp=atan2d(cosd(aposv),sind(aposv));%Az aposv em coordenadas polares
if aposv_cp >= 0
    adosv=aposv_cp-180;
else
    adosv=aposv_cp+180
end

thetaAB=acosd(cosd(rAB)*cosd(adosv)+sind(rAB)*sind(adosv));
end
```

FIGURA 3.4: Transformação dos ângulos do rumo de  $A$  para  $B$  e do ângulo do vento para coordenadas polares.

#### 3.2.3 Cálculos Iniciais para duas Pernadas

Na realização dos cálculos para duas pernadas prossegue-se à criação de um novo ponto de passagem  $P$  que representa o ponto onde vai ser realizada a guinada da primeira pernada para a segunda. Nesta fase da resolução todos os pontos da malha base são considerados como ponto  $P$ , sendo que a distância e os rumos vão ser calculados do ponto  $A$  até ao  $P$  e do ponto  $P$  até ao ponto  $B$ . A Figura 3.5 apresenta como foi realizada a programação para o cálculo das novas variáveis obtidas, sendo elas: distância  $A - P$  ( $dAP$ ), distância  $P - B$  ( $dPB$ ), rumo  $A - P$  ( $rAP$ ) e rumo  $P - B$  ( $rPB$ ), todas em forma de matriz.

```
function [dAP,rAP, dPB, rPB]=trajecto1P(Xbase,Ybase,A,B)
%rAB e rPB rumos das pernadas relativamente a LESTE
%
[m,n] = size(Xbase); % Determina as dimensões da matriz Xbase
Amx=ones(m,n)*A(1); % Matriz cujas entradas são a 1ª coordenada de A
Amy=ones(m,n)*A(2); % Matriz cujas entradas são a 2ª coordenada de A
Bmx=ones(m,n)*B(1); % Matriz cujas entradas são a 1ª coordenada de B
Bmy=ones(m,n)*B(2); % Matriz cujas entradas são a 1ª coordenada de B
%
% Calcular as distâncias do ponto A ao ponto P
dAP=sqrt((Xbase-Amx).^2+(Ybase-Amy).^2);
% Calcular as distâncias do ponto P ao ponto B
dPB=sqrt((Xbase-Bmx).^2+(Ybase-Bmy).^2);
%
%Calcular o rumo do ponto A ao ponto P
rAP=atan2d(Ybase-Amy,Xbase-Amx);
%Calcular o rumo do ponto A ao ponto P
rPB=atan2d(Bmy-Ybase,Bmx-Xbase);
end
```

FIGURA 3.5: Programação para obtenção das distâncias e rumos do ponto  $A$  para  $P$  e de  $P$  para  $B$ .

Como podemos verificar na Figura 3.6 é realizada novamente uma transformação dos ângulos dos rumos ( $rAP$  e  $rPB$ ) para coordenadas polares ( $thetaAP$  e  $thetaPB$ ), usando a mesma metodologia apresenta acima (Sub-subcapítulo 3.2.2).

```
function [thetaAP,thetaPB]=theta1P(rAP,rPB,aposv)
%rAP e rPB rumos em coordenadas polares [-180 a +180]
%aposv-Azimuth Para Onde Sopra o Vento
%adosv-Azimuth De Onde Sopra o Vento
% thetaAP,thetaPB Ângulos de entrada nos diagramas polares (das pernadas)

[m,n] = size(rAP); % Determina as dimensões da matriz rAP

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Conversão de "aposv" (0[N] a 360) em "adosv" [-180 0[E] +180[
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

aposv_cp=atan2d(cosd(aposv),sind(aposv));%Az aposv em coordenadas polares
if aposv_cp >= 0
    adosv=aposv_cp-180;
else
    adosv=aposv_cp+180;
end

adosv=ones(m,n)*adosv;

thetaAP=acosd(cosd(rAP).*cosd(adosv)+sind(rAP).*sind(adosv));
thetaPB=acosd(cosd(rPB).*cosd(adosv)+sind(rPB).*sind(adosv));

end
```

FIGURA 3.6: Transformação dos ângulos do rumo de  $A$  para  $P$  e de  $P$  para  $B$  e do ângulo do vento para coordenadas polares.



### 3.2.4 Geração do Diagrama Polar

Após serem obtidos os resultados das distâncias e rumos entre todas as trajetórias  $A - P - B$  possíveis, é necessário proceder ao cálculo do tempo necessário para as realizar. A velocidade do veleiro é obtida através do diagrama polar programado no *MATLAB*®.

Este programa possui quatro diagramas polares programados que podem ser visualizados no *Apêndice B*, sendo que dois deles, o diagrama polar do N.R.P. (Navio da República Portuguesa) “Sagres” (*gera\_diag\_polar\_sagres*), e o do veleiro do tipo “Albin 79” (*gera\_diag\_polar\_albin*) foram gerados através da leitura direta do diagrama polar destes (Figura 3.7). O diagrama polar do veleiro *Alba* foi obtido através da leitura direta da tabela de velocidades presente no *Anexo II*. Finalmente, o diagrama polar dos veleiros do tipo “SB 20” (*gera\_diag\_polar\_sb20*), recentemente obtidos pela EN para competição, foi gerado através de uma tabela de velocidades presente na sua ficha técnica que pode ser visualizada no *Anexo I*.

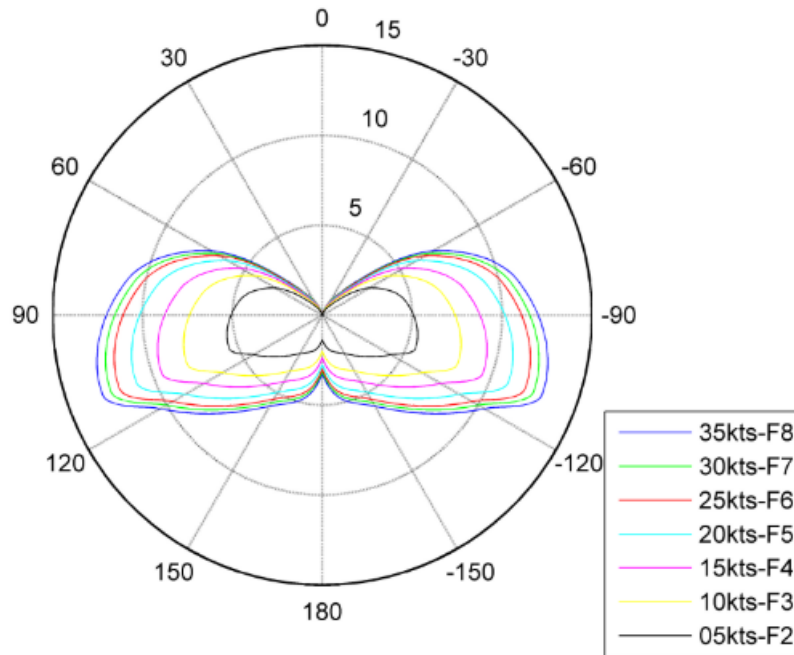


FIGURA 3.7: Diagrama polar em forma gráfica do N.R.P. “Sagres” (imagem adaptada de da Mota, 2014).

Como exemplo de transformação nesta explicação vai ser usado o diagrama polar do N.R.P. “Sagres” apresentado na Figura 3.7. Este diagrama polar foi gerado experimentalmente na dissertação de mestrado de da Mota, 2014, onde “foram inseridas 1357 entradas horárias de dados relativos ao comportamento do navio (a navegar somente à vela) e das condições ambientais que se faziam sentir”(da Mota, 2014).

Inicialmente, é passada toda a informação recolhida do gráfico construindo um polinómio por partes (PP), visto que como podemos verificar na Figura 3.7 há

várias velocidades do vento que não são apresentados no gráfico, daí ser necessário construir o dito PP, para mais tarde através de interpolações do tipo “*spline*”<sup>13</sup> ser obtida a velocidade do veleiro necessária para os cálculos. O formato final do subprograma criado para o N.R.P. “Sagres” pode ser visualizado na Figura 3.8.

```
function [diag_polar, x, y]=gera_diag_polar_sagres(vento)
% Edifica o pp (piecewise polynomial) correspondente ao vento verdadeiro,
% diag_polar, que será usado nos cálculos massivos das velocidades
% Sagres POLAR
%
%Ângulos de entrada do vento
x=[0, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180];
%Velocidades do veleiro consoante velocidade do vento
if 0 <= vento & vento < 5
    y5=[0,0.2,0.8,2.5,4.5,5.0,5.8,4.2,2.5,2.3,2.0,1.7];
    y=y5*vento/5;
elseif vento < 10
    y5=[0,0.2,0.8,2.5,4.5,5.0,5.8,4.2,3.4,3.0,2.5,1.7];
    y10=[0,0.3,1.0,4.6,6.3,7.1,7.7,6.3,4.8,3.5,2.5,2.2];
    y=y5+(vento-5)*(y10-y5)/5;
elseif vento < 15
    y10=[0,0.3,1.0,4.6,6.3,7.1,7.7,6.3,4.8,3.5,3.2,2.2];
    y15=[0,0.4,1.2,5.0,7.4,8.5,9.2,7.5,6.0,4.2,3.5,2.5];
    y=y10+(vento-10)*(y15-y10)/5;
elseif vento < 20
    y15=[0,0.4,1.2,5.0,7.4,8.5,9.2,7.5,6.0,4.2,3.5,2.5];
    y20=[0,0.5,1.4,6.0,8.5,10.1,11.0,8.5,6.5,5.0,4.0,2.9];
    y=y15+(vento-15)*(y20-y15)/5;
elseif vento < 25
    y20=[0,0.5,1.4,6.0,8.5,10.1,11.0,8.5,6.5,5.0,4.0,2.9];
    y25=[0,0.6,1.6,6.4,9.5,11.0,12.1,9.6,7.3,5.4,4.5,3.2];
    y=y20+(vento-20)*(y25-y20)/5;
elseif vento < 30
    y25=[0,0.6,1.6,6.4,9.5,11.0,12.1,9.6,7.3,5.4,4.5,3.2];
    y30=[0,0.7,1.7,6.6,10.0,11.4,12.5,10.0,7.4,5.6,4.8,3.3];
    y=y25+(vento-25)*(y30-y25)/5;
elseif vento < 35
    y30=[0,0.7,1.7,6.6,10.0,11.4,12.5,10.0,7.4,5.8,4.8,3.3];
    y35=[0,0.8,1.8,7.2,10.4,12.1,12.7,10.4,7.5,6.0,5.0,3.5];
    y=y30+(vento-30)*(y35-y30)/5;
else
    return
end
%Interpolação final que fornece a velocidade do veleiro consoante a
%velocidade do vento
diag_polar = spline(x,y);
end
```

FIGURA 3.8: Programação do diagrama polar do N.R.P. “Sagres”.

<sup>13</sup>*spline*: Realiza a construção da forma do polinómio por partes utilizando um interpolador cúbico *ppval* (MathWorks, 2005).

### 3.2.5 Cálculo da Velocidade e Tempo Necessário para Realização do Trajeto

Para proceder aos cálculos de velocidade para cada trajeto  $A - P - B$ , continuando com o exemplo da “Sagres” é chamado o subprograma correspondente a este (*gera\_diag\_polar\_sagres*) que, como podemos verificar na Figura 3.9, realiza os cálculos da velocidade fazendo a interpolação *spline* de acordo com a velocidade do vento para a navegação entre os pontos  $A - P$  ( $vel\_AP$ ), entre  $P - B$  ( $vel\_PB$ ) e a velocidade de uma navegação direta  $A - B$  ( $vel\_AB$ ).

Após serem obtidas as matrizes de velocidade para cada trajeto usando a fórmula de calcular o tempo ( $t$ ) apresentada no subcapítulo 2.1 são obtidas as matrizes de tempo entre cada pernada ( $temp\_AP$  e  $temp\_PB$ ) e o somatório destas ( $temp\_total\_APB$ ), assim como o tempo da navegação direta de  $A$  para  $B$  ( $temp\_total\_AB$ ).

```
% Gera_diag_polar com base nos dados deste tipo de diagrama fornecido pelo
% veleiro. Este diagrama polar está associado à velocidade do vento "vento"
% e será utilizado em todos os cálculos feitos daqui para a frente.
% diag_polar é uma função polinomial (pp) definida por ramos de 0 a 180°.
[diag_polar,x,y]=gera_diag_polar_sagres(vento)
%
%
%
% Cálculo das velocidades em cada pernada e da navegação direta
%Velocidade do ponto A ao ponto P
vel_AP=ppval(diag_polar,thetaAP);
%Velocidade do ponto B ao ponto P
vel_PB=ppval(diag_polar,thetaPB);
%Velocidade do ponto A ao ponto B
vel_AB=ppval(diag_polar,thetaAB)
%
%
% Cálculo dos tempos de percurso em cada pernada
%Tempo de navegação do ponto A ao ponto P
temp_AP=dAP./vel_AP;
%Tempo de navegação do ponto P ao ponto B
temp_PB=dPB./vel_PB;
%
%
% Calculo dos tempos totais de percurso (horas) (distância em milhas
% nauticas e velocidade em nós)
%
%Tempo total navegação com duas pernadas
tempo_total_APB=temp_AP+temp_PB
```

FIGURA 3.9: Cálculo das velocidades e tempo necessário entre os ponto  $A - P - B$ .

### 3.2.6 Preparação para a Utilização da Técnica de Simulação de Recozimento

Após serem obtidos os tempos de todos os trajetos  $A - P - B$  possíveis é realizada a organização desses tempos por ordem crescente ( $tempo\_total\_APB\_C$ ), ou seja o valor do tempo que aparece em primeiro lugar na matriz corresponde ao melhor tempo e o último ao pior, sendo que as coordenadas correspondentes a cada ponto  $P$  nesses trajetos ( $Xbase\_C$  e  $Ybase\_C$ ) também organizado consoante o tempo, como podemos verificar na Figura 3.10.

```
%% INICIO SIMULATING ANNEALING
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colocar os valores do tempo total por ordem crescente através da função
% "sort" e transformar a matriz num vetor.
%Xbase, Ybase, tempo_total_APB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[tempo_total_APB_C id]=sort(tempo_total_APB(:));
%
%
%Colocar os valores das coordenadas de acordo com os valores do tempo,
%transformando a matriz num vetor
Xbase_C=Xbase(id);
Ybase_C=Ybase(id);
%
```

FIGURA 3.10: Organização dos trajetos e das coordenadas correspondentes por ordem crescente de tempo.

De seguida são memorizados os primeiros  $4 \times \sqrt{linhas \times colunas}$  ( $val$ ) melhores valores de tempo ( $tempo\_total\_APB\_1$ ) e as correspondentes coordenadas ( $Xvector\_1$  e  $Yvector\_1$ ), sendo guardado o melhor resultado de tempo nesta fase do script ( $melhor\_tempo$ ,  $melhor\_X$  e  $melhor\_Y$ ), ver Figura 3.11.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Dimensão("val") do espaço a minimizar 4*sqrt(m*n)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[l,c]=size(tempo_total_APB);
%
val=round(4*sqrt(1*c));
%
%Guardar "val" resultados
tempo_total_APB_1=tempo_total_APB_C(1:val);
Xvector_1=Xbase_C(1:val);
Yvector_1=Ybase_C(1:val);
%
%
%Guardar o melhor resultado
melhor_tempo=tempo_total_APB_1(1);
melhor_X=Xvector_1(1);
melhor_Y=Yvector_1(1);
```

FIGURA 3.11: Guardar os melhores  $4 \times \sqrt{linhas \times colunas}$  e o melhor resultado na altura.

Como foi explicado no Sub-subcapítulo 2.4.4 a técnica de SR realiza uma perturbação nos valores iniciais, neste caso a perturbação inicial (*amp\_b*) vai corresponder a  $resolucao/2$  e o ciclo (*epocas*) vai corresponder a um número de iterações que pode ser devidamente ajustado pelo utilizador. Nos cálculos efetuados foram consideradas suficientes 100 iterações.

#### 3.2.7 Início do ciclo da Simulação de Recozimento

Chegando ao ciclo em si, inicialmente os resultados de menor tempo obtidos anteriormente (*melhor\_tempo*, *melhor\_X* e *melhor\_Y*) são guardados com um novo nome (*total\_tempo\_tmp*, *Xvector\_tmp* e *Yvector\_tmp*) para ser guardado de ciclo para ciclo do melhor resultado da geração anterior. Também é necessário fazer com que as perturbações (*amp\_b*) sejam cada vez menores de um ciclo para outro, e divididas por 1.01, ver Figura 3.12.

```
for i1=1:epocas
    %Guardar o melhor tempo da geração anterior e as correspondentes
    %coordenadas
    total_tempo_tmp=melhor_tempo;
    Xvector_tmp=melhor_X;
    Yvector_tmp=melhor_Y;
    %
    %Diminuição da amplitude perturbadora ao longo das iterações
    amp_b=amp_b/1.01;
    %
    n_min=-amp_b;
    n_max=amp_b;
```

FIGURA 3.12: Guarda do melhor resultado da geração anterior e diminuição do espaço de perturbação.

Após isto são criados os valores máximos (*n\_max*) e mínimos (*n\_min*) entre os quais são perturbados os valores de cada geração. A perturbação é realizada através de vetores (*Xrandom* e *Yrandom*) que aleatoriamente vão modificar a posição dos pontos existentes *P* (*Xvector\_1* e *Yvector\_1*) (Figura 3.13).

```
%Criação dos vetores que vão fazer a perturbação
Xrandom=n_min+(n_max-n_min)*rand(val,1); %amplitude corrente X
Yrandom=n_min+(n_max-n_min)*rand(val,1); %amplitude corrente Y
%Realização da perturbação nos resultados existentes
%
Xvector_1=plus(Xvector_1,Xrandom);
Yvector_1=plus(Yvector_1,Yrandom);
```

FIGURA 3.13: Criação dos vetores perturbantes e a realização da perturbação.

Posteriormente à perturbação realizada nas posições existentes da geração anterior os cálculos são refeitos para essas novas posições e é guardado o melhor representante da geração anterior, como podemos observar na Figura 3.14.

```
%Calcular os novos resultados
%
[dAP_1,rAP_1, dPB_1, rPB_1]=trajectoiP(Xvector_1,Yvector_1,A,B);
[thetaAP_1,thetaPB_1]=theta1P(rAP_1,rPB_1,aposv);
vel_AP_1=ppval(diag_polar,thetaAP_1);
vel_PB_1=ppval(diag_polar,thetaPB_1);
temp_AP_1=dAP_1./vel_AP_1;
temp_PB_1=dPB_1./vel_PB_1;
tempo_total_APB_1=temp_AP_1+temp_PB_1;
% Preservar o melhor representante anterior
[tempo_total_APB_1 id]=sort(tempo_total_APB_1);
Xvector_1=Xvector_1(id);
Yvector_1=Yvector_1(id);
```

FIGURA 3.14: Realização de novos cálculos após a perturbação das posições.

Finalmente, é substituído o último valor desta geração pelo primeiro da geração anterior e é novamente realizada a organização dentro da matriz por ordem crescente de tempo de trajeto, sendo novamente guardado o melhor resultado desta geração e as suas coordenadas (*melhor\_tempo*, *melhor\_X* e *melhor\_Y*). No final de cada geração, também é criada uma variável (*tempo\_min*) que permite, como vai ser demonstrado mais adiante, mostrar a evolução do tempo mínimo para a realização do trajeto de geração para geração, como é mostrado na Figura 3.15.

```
%Substituir o ultimo valor desta geração pelo primeiro da geração
%anterior e fazer o "sort"
tempo_total_APB_1(end)=total_tempo_tmp;
Xvector_1(end)=Xvector_tmp;
Yvector_1(end)=Yvector_tmp;
%
[tempo_total_APB_1 id]=sort(tempo_total_APB_1);
Xvector_1=Xvector_1(id);
Yvector_1=Yvector_1(id);
%
%
%Guardar o melhor resultado desta geração para ser usado na geração
%seguinte
melhor_tempo=tempo_total_APB_1(1);
melhor_X=Xvector_1(1);
melhor_Y=Yvector_1(1);
%
tempo_min=[tempo_min, melhor_tempo]; %Criado para ver a evolução do melhor
%tempo de geração para geração
```

FIGURA 3.15: Fase final de cada geração e a criação da variável para geração de um gráfico (*tempo\_min*).

### 3.2.8 Demonstração de Resultados em Forma Gráfica

Finalizando o *script* e já obtida a trajetória ótima para as condições impostas é necessário demonstrar esses resultados numa forma gráfica. Assim sendo, foram criados no *software MATLAB*<sup>®</sup> dois gráficos que são expostos após todos os

cálculos estarem concluídos, o primeiro para verificar a evolução do tempo mínimo com o decorrer dos ciclos (Figura 3.16) e o segundo que representa os trajetos  $A-B$  e o trajeto ótimo  $A-P-B$  (Figura 3.17). O código completo da criação dos gráficos e a forma como são gravados para uso no Capítulo 4 está contido no Apêndice C.

```
#####
%Apresentação dos resultados em forma de gráficos
#####
%Plot para ver a evolução do tempo mínimo
figure(1)
plot(tempo_min, '-k', 'color', 'red', 'LineWidth', 1.5);
set(gca, 'FontSize', 10, 'FontWeight', 'bold', 'LineWidth', 2.0, 'box', 'on') ;
a1=xlabel(['\it Ciclos ', '\rm [n°]']); set(a1, 'FontSize', 12, 'FontWeight'...
, 'bold');
a2=ylabel(['\it T', '\rm [h]']); set(a2, 'FontSize', 12, 'FontWeight', 'bold');
a3=title('Evolução do tempo mínimo com o decorrer dos ciclos');
set(a3, 'FontSize', 12, 'FontWeight', 'bold')
grid on
```

FIGURA 3.16: Criação do gráfico de evolução do menor tempo com o decorrer das épocas.

```
%Plot dos trajectos A-B e A-P-B
figure(2)
pontos=plot(A(1), A(2), 'x', melhor_X(1), melhor_Y(1), 'o', B(1), B(2), 's');
set(pontos, 'MarkerSize', 8, 'MarkerEdgeColor', 'k', 'LineWidth', 1.5);
set(gca, 'FontSize', 10, 'FontWeight', 'bold', 'LineWidth', 2.0, 'box', 'on') ;
b1=xlabel('Coordenada X'); set(b1, 'FontSize', 12, 'FontWeight', 'bold')
b2=ylabel('Coordenada Y'); set(b2, 'FontSize', 12, 'FontWeight', 'bold')
b3=title('Trajetos A-B e A-P-B'); set(b3, 'FontSize', 12, 'FontWeight', 'bold')
grid on
hold on
quiver1=quiver(A(1), A(2), B(1)-A(1), B(2)-A(2), 0);
set(quiver1, 'MarkerSize', 8, 'MarkerEdgeColor', 'k', 'LineWidth', 1.5);
hold on
plot([A(1) melhor_X(1)], [A(2) melhor_Y(1)], '--k', 'LineWidth', 1.5)
quiver(A(1), A(2), melhor_X(1)-A(1), melhor_Y(1)-A(2), 0)
hold on
plot([melhor_X(1) B(1)], [melhor_Y(1) B(2)], '--k', 'LineWidth', 1.5)
quiver(melhor_X(1), melhor_Y(1), B(1)-melhor_X(1), B(2)-melhor_Y(1), 0)
axis equal
hold on
#####
% VENTO
#####
px=sind(aposv); py=cosd(aposv);
quiver(Xbase, Ybase, 0*Xbase+px, 0*Ybase+py, 'r')
legend('A', 'P', 'B')
```

FIGURA 3.17: Criação do gráfico com os trajeto  $A-B$  e a trajetória ótima  $A-P-B$ .

### 3.2.9 Resumo da Abordagem do Problema

Resumindo, este *script* foi desenvolvido para resolução do problema da Instância A1 obtendo assim um trajeto ótimo com duas pernadas. A continuação da resolução desta instância com mais pernadas fica deferida para próximos desenvolvimentos devido à necessidade de consideração das perdas de velocidade associadas

às mudanças de bordo que têm de ser traduzidas em forma de cálculo. Na Figura 3.18 está resumido o *script* realizado, em forma de fluxograma, seguindo-se do pseudocódigo correspondente na Figura 3.19, para uma mais fácil percepção dos passos de resolução utilizados e implementados.

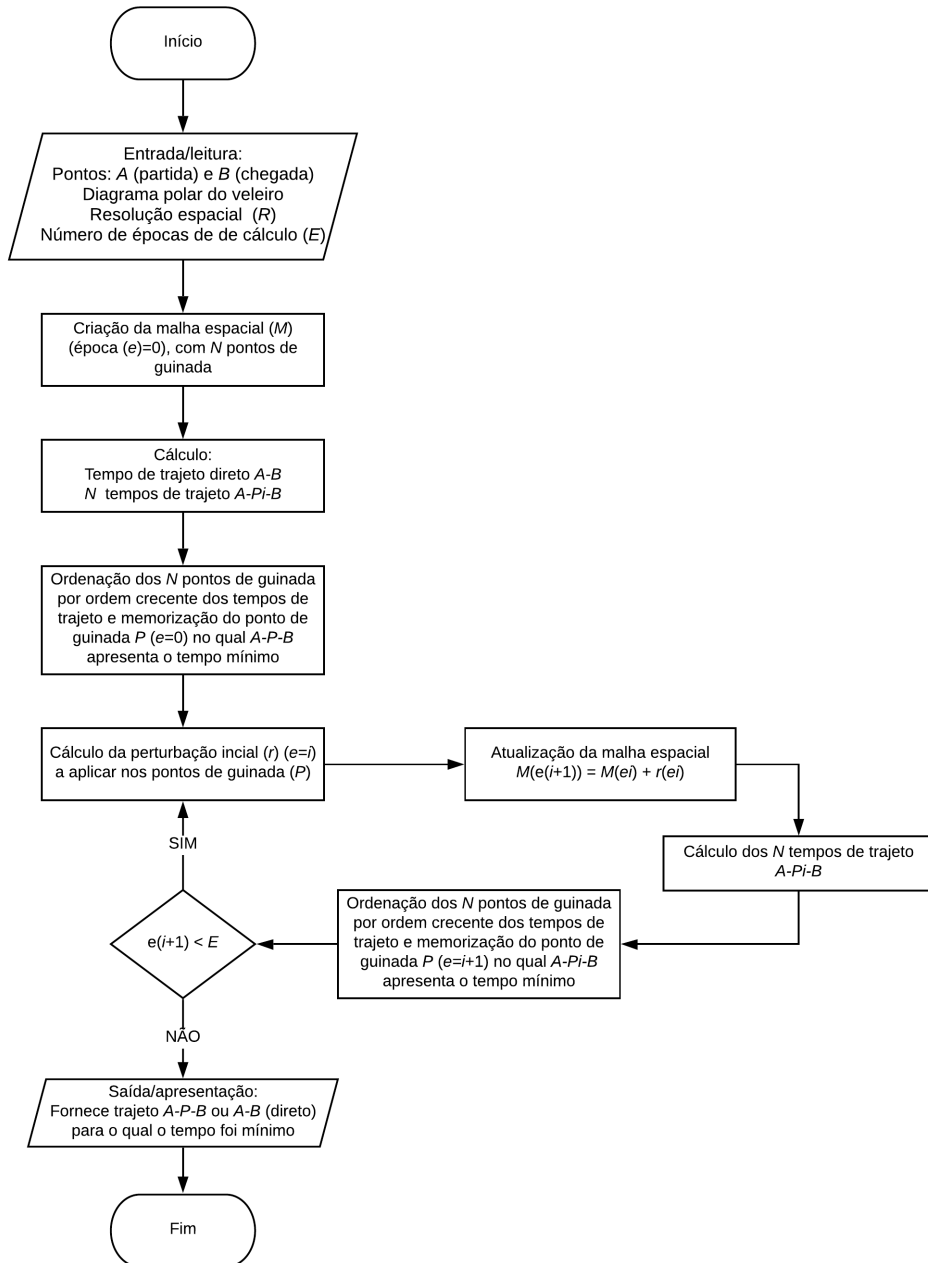


FIGURA 3.18: Fluxograma do *script* realizado.



### 3.2. Abordagem do Problema (Instância A1)

---

**Algoritmo :** Trajetória ótima entre A e B utilizando a Simulação de Recozimento

*A* - Ponto de partida

*B* - Ponto de chegada

*P* - Ponto de guinada

*D* - Diagrama polar do veleiro

*R* - Resolução espacial

*E* - Número de épocas de cálculo

*e* - Época

*M* - Malha base

*r* - Perturbação inicial

**INÍCIO**

**LER**

1. *A*
2. *B*
3. *D*
4. *R*
5. *E*

**FAZER**

6. Criação de *M* com *N* pontos de guinada
7. Cálculo do trajeto direto *A-B*
8. Cálculo de *N* trajetos *A-Pi-B*
9. Ordenação crescente de tempo dos *N* pontos *P*
10. Memorização de *P* que apresenta o tempo mínimo *A-P-B*
11. Cálculo de *r* ( $e=i$ ) a aplicar em *P*
12. Atualização de *M* -  $M(e_{i+1}) = M(e_i) + r(e_i)$
13. Calcular *N* tempos de trajeto *A-P-B*
14. Ordenação crescente de tempo dos *N* pontos *P*
15. Memorização de *P* ( $e=i+1$ ) que apresenta o tempo mínimo *A-P-B*
16. Enquanto  $e(i+1) < E$  executar 11, 12, 13, 14 e 15

**FORNECER**

17. Trajeto *A-P-B* ou *A-B* para o qual o tempo é mínimo

**FIM**

FIGURA 3.19: Pseudocódigo do *script* realizado.



# Capítulo 4

## Simulações numéricas

Neste capítulo vão ser apresentados e discutidos os resultados de várias simulações numéricas realizadas utilizando o *script* desenvolvido para resolução do problema abordado no capítulo anterior. Inicialmente será justificada a razão pela qual o número de épocas utilizadas foram apenas 100. Irão também ser considerados diferentes cenários para testar os algoritmos desenvolvidos.

### 4.1 Geração do Gráfico de Evolução do Tempo Mínimo Obtido com o Decorrer das Épocas

Após a realização de vários testes para diferentes cenários e diferentes veículos notou-se que a nível de cálculo verificava-se que a partir de um determinado número de épocas, ao utilizar a técnica de Simulação de Recozimento, não se verificava mudanças significativas no tempo mínimo para realização do trajeto, como podemos verificar na Figura 4.1, onde foram utilizados 1000 épocas. Utilizando o mesmo cenário foi realizada outra simulação, desta vez com 100 épocas (Figura 4.2), obtendo-se um resultado idêntico.

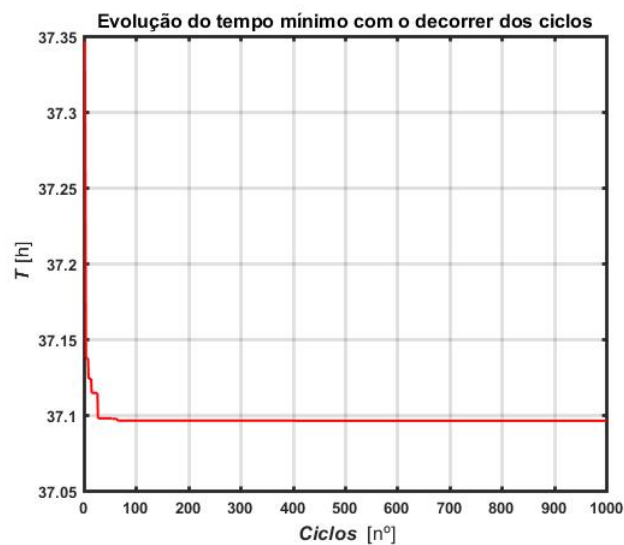


FIGURA 4.1: Evolução do tempo mínimo obtido com o decorrer de 1000 ciclos.

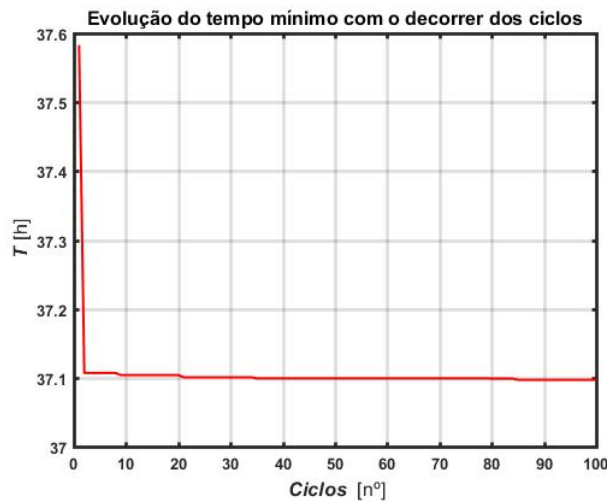


FIGURA 4.2: Evolução do tempo mínimo obtido com o decorrer de 100 ciclos.

Posto isto, para não sobrecarregar desnecessariamente o programa, o que leva a um aumento do tempo necessário para obtenção do resultado, considerou-se suficiente que o número de ciclos para todos os testes realizados fosse igual a 100.

## 4.2 Geração do Melhor Trajeto para Diferentes Veleiros, Usando o Mesmo Cenário

Neste subcapítulo é realizada uma simulação onde alguns dos veleiros cujos diagramas polares foram programados e apresentados no Sub-subcapítulo 3.2.4, nomeadamente os veleiros do tipo “Albin 79”, os veleiro “SB 20” e o N.R.P. “Sagres”, descrevem trajetos contra o vento com os parâmetros mostrados na Tabela 4.1.

Posição inicial ( $A$ )	Posição final ( $B$ )	Velocidade do vento ( $vento$ )	Direção para a qual sopra o vento ( $aposv$ )	Variável de referência ( $passos$ )
(0,0)	(100,0)	13 nós	270	5

TABELA 4.1: Parâmetros iniciais para a navegação contra o vento.

Após serem inseridos os valores de cenário apresentados acima o programa apresentou os seguintes resultados:

#### 4.2. Geração do Melhor Trajeto para Diferentes Veleiros, Usando o Mesmo Cenário

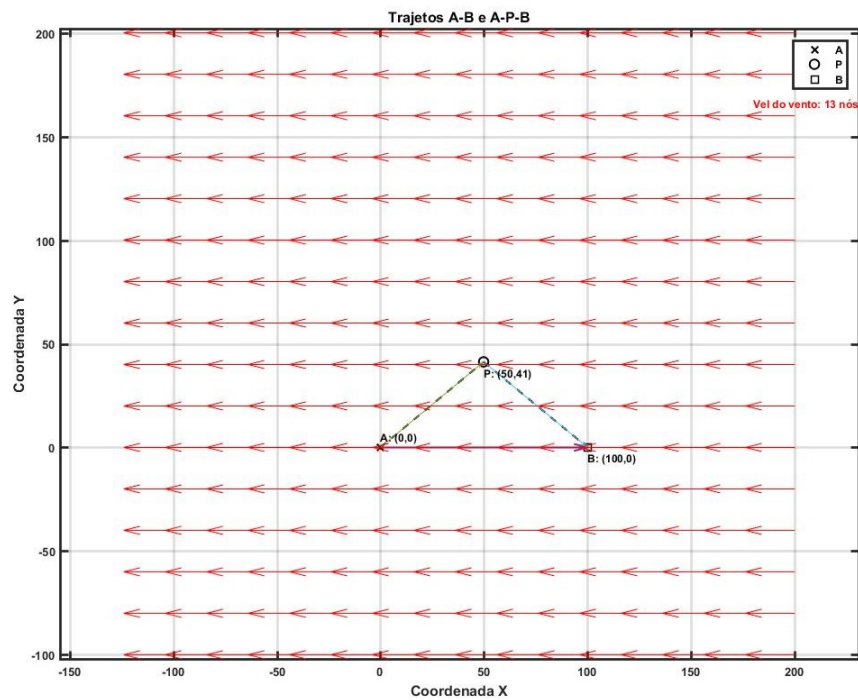


FIGURA 4.3: Navegação contra o vento do veleiro “Albin 79”, com uma velocidade do vento de 13 nós.

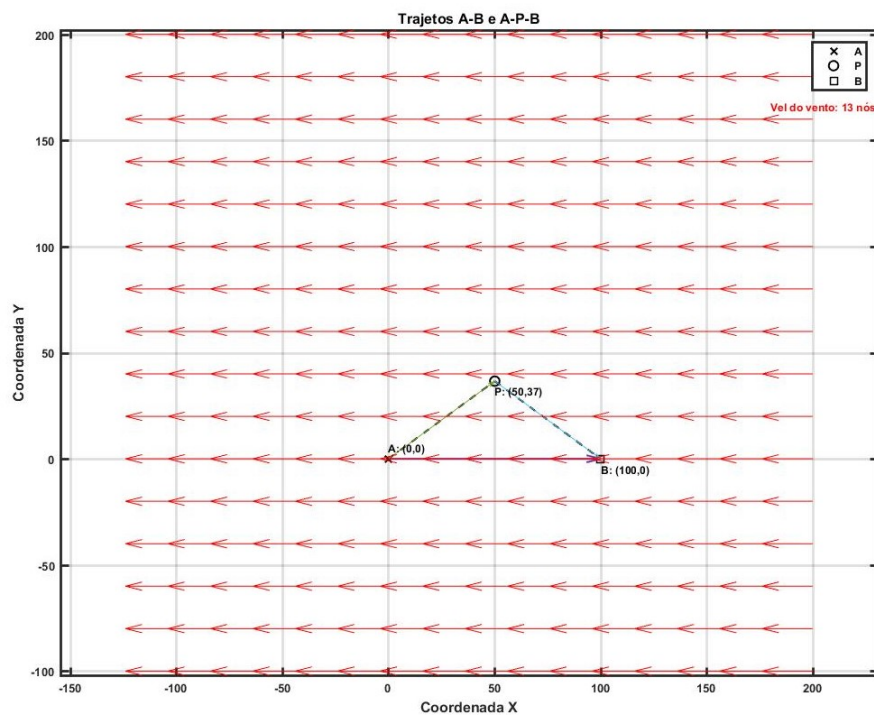


FIGURA 4.4: Navegação contra o vento do veleiro “SB 20”, com uma velocidade do vento de 13 nós.

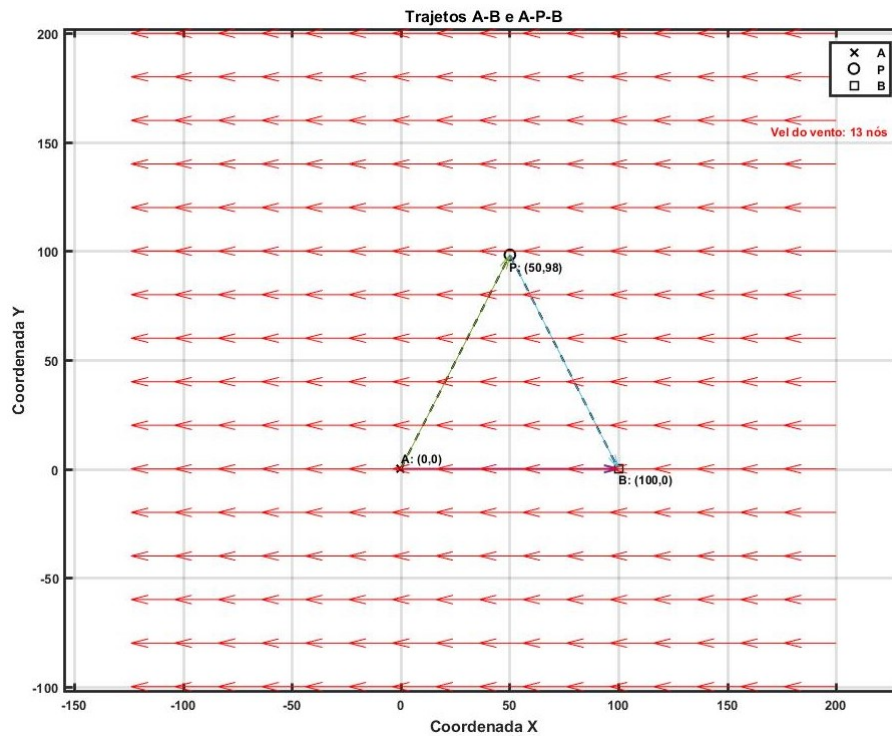


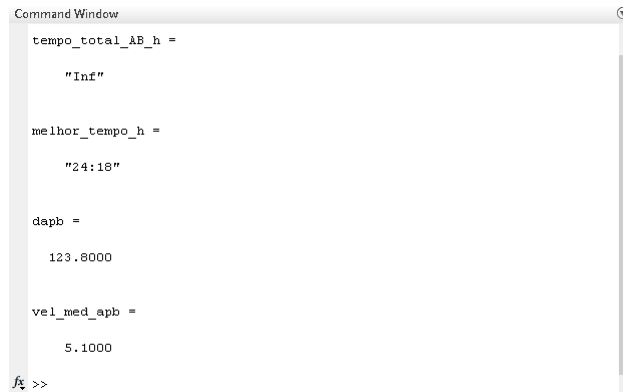
FIGURA 4.5: Navegação contra o vento do N.R.P. "Sagres" , com uma velocidade do vento de 13 nós.

Como podemos verificar nas Figuras 4.3, 4.4 e 4.5, o objetivo desta simulação é realizar uma navegação com uma distância de cem milhas náuticas entre ponto inicial  $A$  e o ponto final  $B$ , contra o vento, em que a velocidade do vento é de 13 nós. Sabemos com base no subcapítulo 2.3.7 e a partir da observação dos diagramas polares dos veleiros presentes no Apêndice B que a navegação diretamente contra o vento é impossível. Assim sendo a navegação é realizada em duas pernadas, onde em todos os casos o momento de guinada é no ponto com a mesma coordenada  $x$ , concluindo assim que as pernadas para cada veleiro são simétricas.

Veleiro	Distância $A-P-B$ (mi)	Tempo $A-P-B$ (hh:mm)	Velocidade média $A-P-B$ (nós)
"Albin 79"	129.2	26:51	4.80
"SB 20"	124.0	24:18	5.10
NRP "Sagres"	220.5	40:20	5.5

TABELA 4.2: Distâncias a percorrer, tempo necessário e velocidades dos veleiros na navegação contra o vento, com uma velocidade do vento de 13 nós.

A nível de resultados apresentados na Tabela 4.2, estes também são obtidos através do *script*, mas não são visíveis no gráfico, estando apresentados na janela de comandos do *MATLAB*®, como se pode verificar na Figura 4.6 o exemplo dos resultados obtidos para o veleiro "SB 20" .



```
Command Window

tempo_total_AB_h =

    "Inf"

melhor_tempo_h =

    "24:18"

dapb =

    123.8000

vel_med_apb =

    5.1000

fx >>
```

FIGURA 4.6: Janela de comando do *MATLAB*<sup>®</sup> com os resultados obtidos para o veleiro “SB 20”.

Na Figura 4.6 são apresentados os resultados referentes:

- Ao tempo de navegação direta de *A* para *B* (*tempo\_total\_AB\_h*), que apresenta um valor “Inf” porque a navegação direta é impossível e a demora dessa navegação matematicamente corresponde a infinito;
- Ao tempo o trajeto ótimo (*melhor\_tempo\_h*) em horas e minutos;
- À distância total necessária de percorrer ao realizar o trajeto com duas pernadas (*dapb*);
- E, por fim, à velocidade média à qual o veleiro realiza o trajeto ótimo de duas pernadas (*vel\_med\_apb*).

No que toca à interpretação dos resultados obtidos podemos dizer que estão de acordo com as conclusões apresentadas no Sub-subcapítulo 2.3.7. Mais detalhadamente, se observarmos as Figuras 4.3 e 4.4 podemos notar uma semelhança nos seus trajetos, pois tratam-se de dois veleiros de pano latino com características semelhantes, a diferença está contida no facto de que os veleiros do tipo “Albin 79” são de cruzeiro e os do tipo “SB 20” são utilizados em competição, daí haver uma diferença na velocidade média e correspondentemente no tempo necessário para a realização do trajeto observados na Tabela 4.2.

No que toca ao N.R.P. “Sagres”, sendo a maioria do seu velame<sup>14</sup> constituído por pano redondo e apresentando dimensões muito superiores aos outros dois veleiros, este apresenta maior dificuldade na navegação contra o vento (como se pode observar no diagrama polar presente no Apêndice B.2). Posto isto, a trajetória ótima apresenta uma grande “abertura” do rumo em relação ao vento, obrigando assim a percorrer mais milhas e demorar mais tempo, apesar de apresentar uma velocidade média superior ao restantes dois veleiros testados.

Concluindo, estas primeiras simulações conseguem provar que o código criado consegue ser utilizado para diferentes veleiros, desde que seja apresentado o

---

<sup>14</sup>Velame: Conjunto de todas as velas existentes no veleiro (Schmid (1978)).

diagrama polar, em forma de gráfico ou em forma de tabela, apresentando resultados lógicos, como comprovado acima.

### 4.3 Geração do Melhor Trajeto para um Veleiro, Usando Cenários Diferentes

Neste subcapítulo vão ser realizadas simulações numéricas em que o N.R.P. "Sagres" vai ser colocado a navegar em diferentes cenários, seguindo-se a análise dos resultados obtidos.

#### 4.3.1 Navegação à Bolina Cerrada

Esta simulação tem como objetivo calcular a trajetória ótima, com duas pernadas, do N.R.P. "Sagres" para navegação à bolina cerrada<sup>15</sup> fazendo com que o vento entre no veleiro num ângulo de  $45^\circ$  a contar da sua proa. Sendo assim os parâmetros iniciais os presentes na Tabela 4.3.

Posição inicial ( <i>A</i> )	Posição final ( <i>B</i> )	Velocidade do vento ( <i>vento</i> )	Direção para a qual sopra o vento ( <i>apov</i> )	Variável de referência ( <i>passos</i> )
(-10,-50)	(-10,50)	13 nós	225	5

TABELA 4.3: Parâmetros iniciais para a navegação à bolina cerrada.

A trajetória ótima obtida pode ser visualizada na Figura 4.7, sendo que os resultados a nível de tempos e velocidades estão apresentados na Tabela 4.4.

Trajeto	Distância (mi)	Tempo (hh:mm)	Velocidade média (nós)
Direta (A-B)	100	89:17	1.12
2 pernadas (A-P-B)	155.4	28:31	5.4

TABELA 4.4: Distâncias, tempos e velocidades da navegação do N.R.P. "Sagres" à bolina cerrada com uma e duas pernadas.

<sup>15</sup>Bolina cerrada: Navegação à vela em que o vento entra no veleiro pela amura com um ângulo o mais próximo possível do ângulo de onde sopra o vento.



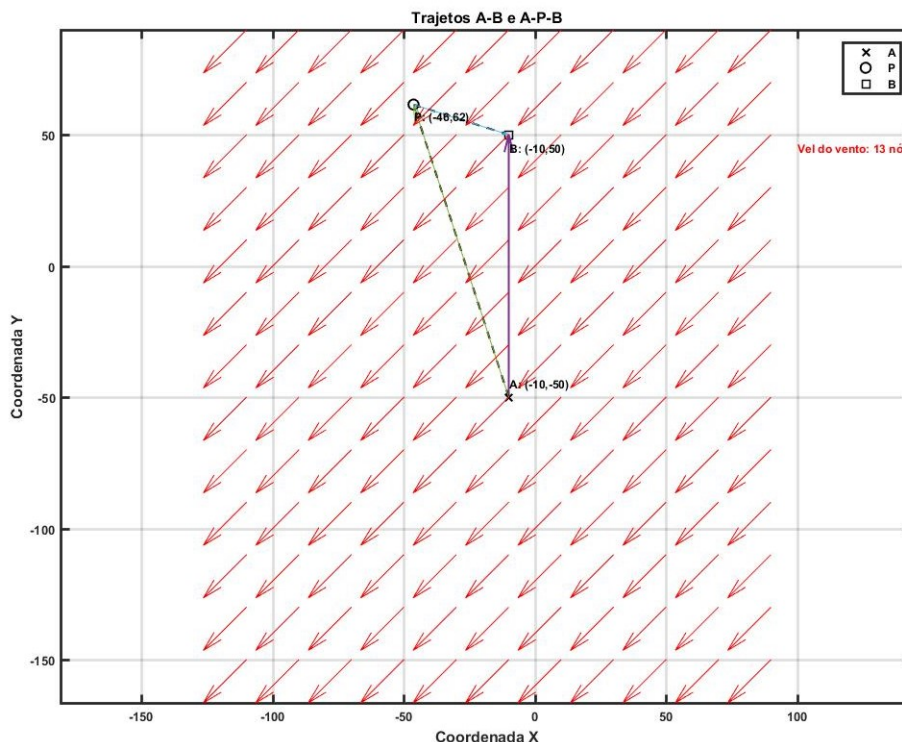


FIGURA 4.7: Trajetória ótima de navegação do N.R.P. "Sagres" à bolina cerrada.

Como podemos verificar na Tabela 4.4 existe uma grande diferença a nível de tempos da navegação direta para a navegação ótima com duas pernadas. Isto deve-se ao facto de a "Sagres", como já foi referido, na maioria do seu velame apresentar pano redondo, o que dificulta a navegação nos setores contra o vento, o que nos é demonstrado no diagrama polar do veleiro (Apêndice B.2). Sendo assim, o programa fornece a trajetória com duas pernadas ótima em que o momento de guinada é o ponto  $P$ , com as coordenadas  $(-46,62)$ . Verifica-se que o veleiro realiza uma navegação em que passa acima o ponto  $B$ , guina e navega em direção a esse ponto, ficando desta forma o tempo necessário para a realização do trajeto aproximadamente três vezes menor.

#### 4.3.2 Navegação com o Vento pelo Través

A simulação numérica que se segue tem o objetivo de encontrar a trajetória ótima para o N.R.P. "Sagres", a navegar com o vento pelo través<sup>16</sup>. Sendo os parâmetros iniciais os observados na Tabela 4.5.

<sup>16</sup>Vento pelo través: Quando o vento entra no veleiro com um ângulo de noventa graus (na perpendicular) em relação à sua proa.

Posição inicial ( $A$ )	Posição final ( $B$ )	Velocidade do vento ( $vento$ )	Direção para a qual sopra o vento ( $apov$ )	Variável de referência ( $passos$ )
$(-35,50)$	$(50,-35)$	13 nós	225	5

TABELA 4.5: Parâmetros iniciais para a navegação com o vento pelo través.

Seguidamente foram obtidos os seguintes resultados:

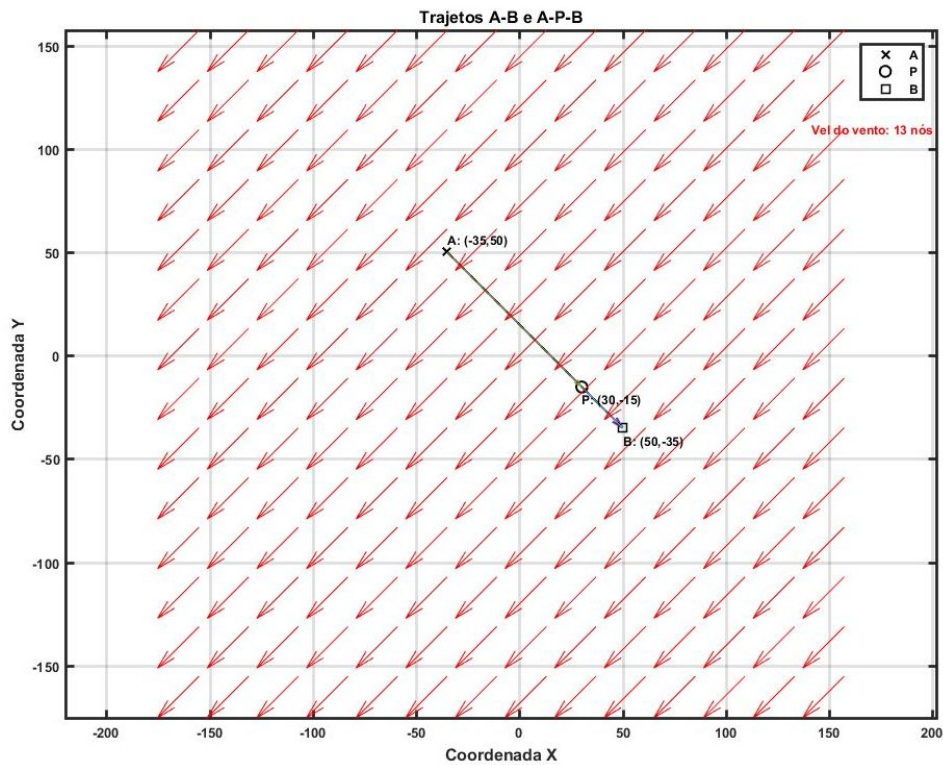


FIGURA 4.8: Trajetória ótima de navegação do N.R.P. "Sagres" com o vento pelo través.

Trajetória	Distância (mi)	Tempo (hh:mm)	Velocidade média (nós)
Direta (A-B)	120.2	15:08	7.94
Duas Pernadas (A-P-B)	120.2	15:08	7.94

TABELA 4.6: Distâncias, tempos e velocidades da navegação do N.R.P. "Sagres" com vento pelo través.

Ao realizar a análise da informação contida na Figura 4.8 e na Tabela 4.6 podemos chegar à conclusão que ambas as trajetórias são iguais. Nesta simulação é demonstrado que a trajetória ótima é a navegação direta do ponto inicial para o final, pois o ponto intermédio  $P$  obtido no cálculo da melhor trajetória está contido

num dos pontos em que passa a trajetória direta. Se observarmos o diagrama polar do veleiro (Apêndice B.2) vemos que quando o vento entra no veleiro com um ângulo de noventa graus a diferença entre a velocidade que o veleiro pratica nessas condições é próxima da velocidade máxima que o veleiro consegue praticar (quando o vento entra no veleiro com um ângulo de  $115^\circ$  a contar da proa). Sendo assim, pensando logicamente, não há razão para o veleiro orçar<sup>17</sup> e navegar com uma velocidade pouco superior numa direção que o afaste em demasia do ponto final, sendo que assim o veleiro navega mais milhas que não são percorridas com uma velocidade superior o suficiente. O que nos leva a concluir que a trajetória ótima neste tipo de cenário é a navegação direta do ponto *A* para o ponto *B*.

#### 4.3.3 Navegação ao Largo

Esta simulação vai apresentar dois cenários. No primeiro cenário o objetivo é testar o programa numa navegação ao largo<sup>18</sup>, em que o vento entra no veleiro com um ângulo de  $115^\circ$ , ângulo pelo qual a velocidade é máxima. Estes cenários apresentam os parâmetros iniciais presentes na Tabela 4.7.

Posição inicial ( <i>A</i> )	Posição final ( <i>B</i> )	Velocidade do vento ( <i>vento</i> )	Direção para a qual sopra o vento ( <i>apov</i> )	Variável de referência ( <i>passos</i> )
(-15,60)	(-15,-65)	13 nós	255	5
(-15,60)	(-15,-65)	13 nós	220	5

TABELA 4.7: Parâmetros iniciais para a navegação ao largo.

Após serem introduzidos os parâmetros iniciais o programa apresentou para o primeiro resultado os resultados apresentados na Tabela 4.8 e Figura 4.9. Os resultados para o segundo cenário estão presentes na Tabela 4.9 e Figura 4.10.

Trajeto	Distância (mi)	Tempo (hh:mm)	Velocidade média (nós)
Direta (A-B)	125	14:32	8.6
Duas Pernadas (A-P-B)	125	14:32	8.6

TABELA 4.8: Distâncias, tempos e velocidades da navegação do N.R.P. "Sagres" com vento a entrar por um ângulo de  $115^\circ$ .

Trajeto	Distância (mi)	Tempo (hh:mm)	Velocidade média (nós)
Direta (A-B)	125	25:15	4.95
Duas Pernadas (A-P-B)	138.2	24:30	5.6

TABELA 4.9: Distâncias, tempos e velocidades da navegação do N.R.P. "Sagres" com vento a entrar por um ângulo de  $140^\circ$ .

<sup>17</sup>Orçar: Afastar a proa do veleiro da linha do vento.

<sup>18</sup>Navegação ao largo: Quando o vento entra no veleiro com um ângulo entre os noventa e os cento e setenta graus em relação à sua proa.

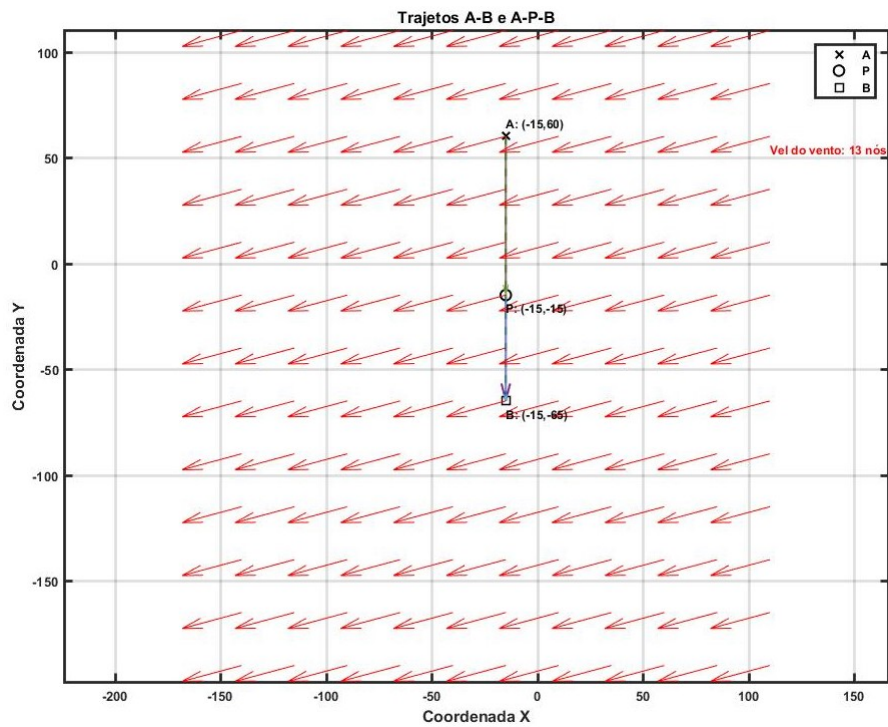


FIGURA 4.9: Trajetória ótima de navegação do N.R.P. "Sagres" com vento a entrar por um ângulo de  $115^\circ$ .

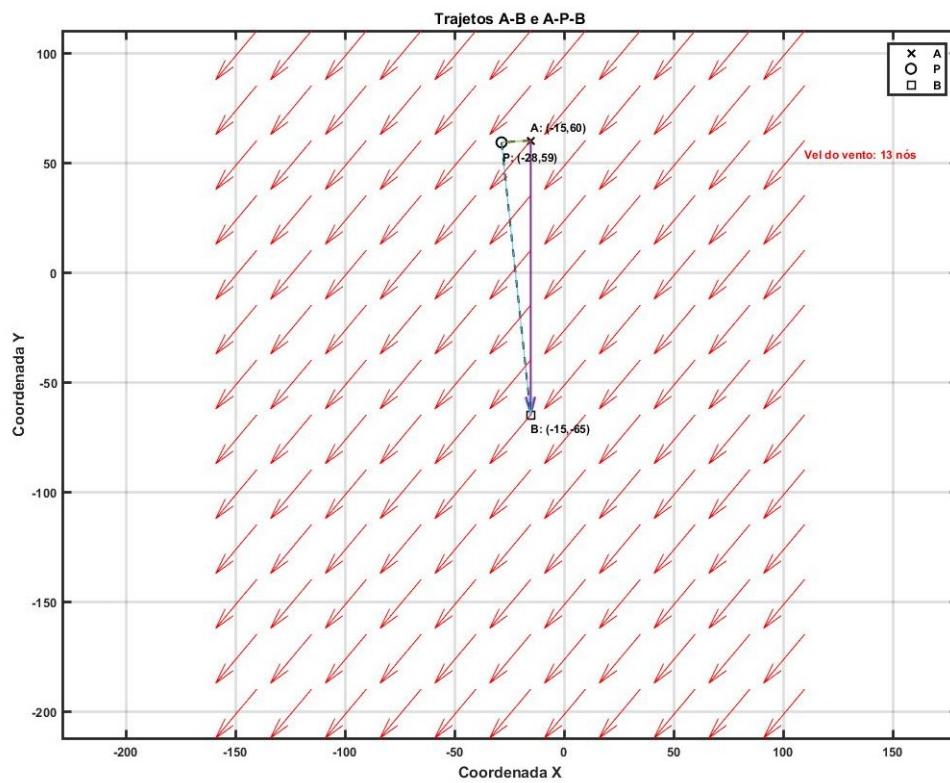


FIGURA 4.10: Trajetória ótima de navegação do N.R.P. "Sagres" com vento a entrar por um ângulo de  $140^\circ$ .

Como pode ser observado no diagrama polar da “Sagres” (Apêndice B.2), é quando o vento entra por um ângulo de  $115^\circ$  que o veleiro atinge a velocidade máxima. Como o primeiro cenário apresenta um vento que entra por um ângulo de  $115^\circ$  é lógico que a trajetória ótima neste caso é a navegação direta do ponto  $A$  para o ponto  $B$ .

No caso do segundo cenário o vento foi alterado para um ângulo de  $140^\circ$  de forma a que continua a ser considerada uma navegação ao largo, mas com um ângulo que se encontra mais perto do setor da popa do veleiro, onde há uma diminuição significativa da velocidade do veleiro. Neste caso a trajetória ótima e a trajetória direta diferem, sendo que a trajetória de duas pernadas ótima inicialmente realiza uma navegação até ao ponto  $P$  com as coordenadas  $(-20;59)$  e após isso navega até  $B$  com uma velocidade superior à da navegação direta.

Concluindo, podemos confirmar o funcionamento do programa a nível de todos os ângulos, mesmo que a diferença entre os ângulos não seja muito grande, sendo que os ângulos pertencem ao mesmo setor.

#### 4.3.4 Navegação com o Vento pela Popa

Finalmente, nesta última simulação vai ser utilizado o cenário que o apresentado no Subcapítulo 4.2, sendo desta vez o vento alterado por forma a entrar no veleiro literalmente pela popa. Os parâmetros iniciais para esta simulação estão contidos na Tabela 4.10.

Posição inicial ( $A$ )	Posição final ( $B$ )	Velocidade do vento ( $vento$ )	Direção para a qual sopra o vento ( $apov$ )	Variável de referência ( $passos$ )
(0,0)	(100,0)	13 nós	090	5

TABELA 4.10: Parâmetros iniciais para a navegação com o vento pela popa.

Os resultados obtidos para este tipo de cenário estão dispostos na Figura 4.11 e na Tabela 4.11.



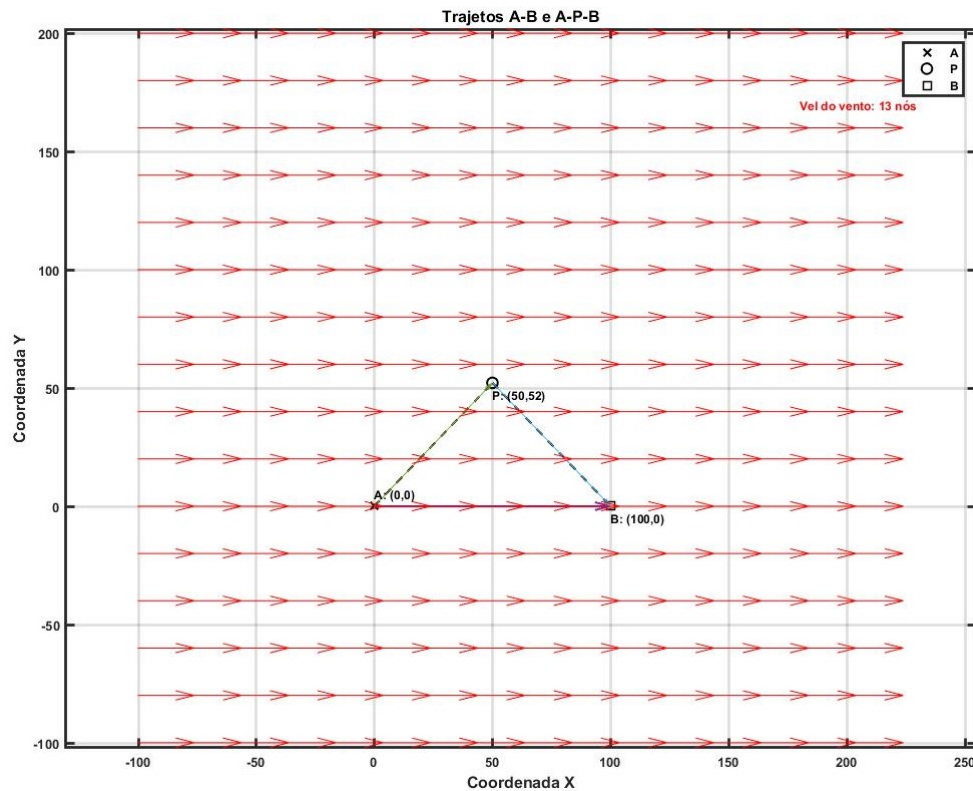


FIGURA 4.11: Trajetória ótima de navegação do N.R.P. "Sagres" com o vento pela popa.

Trajetória	Distância (mi)	Tempo (hh:mm)	Velocidade média (nós)
Direta (A-B)	100	42:01	2.38
Duas Pernadas (A-P-B)	144.5	25:35	5.6

TABELA 4.11: Distâncias, tempos e velocidades da navegação do N.R.P. "Sagres" com vento pela popa.

Analisando os resultados demonstrados na Figura 4.11 e Tabela 4.11 verificamos que há uma diferença entre a trajetória direta e a ótima de duas pernadas. Se observarmos mais uma vez o diagrama polar do veleiro (Apêndice B.2) verificamos que quando o vento passa para os setores da popa é apresentada uma descida a nível da velocidade do veleiro. A trajetória ótima consiste na realização de duas pernadas mais longas mas a uma velocidade superior.

Em relação à simulação apresentada no Subcapítulo 4.2 verificamos que este cenário apresenta os mesmos pontos inicial e final havendo apenas uma alteração da direção de onde sopra o vento. Se verificarmos os resultados apresentados na Tabela 4.4 para o N.R.P. "Sagres" e Tabela 4.11 obtemos tempos para realização do mesmo trajeto bastante diferentes, tudo isto devido ao ângulo pelo qual entra o vento no veleiro.

## 4.4 Conclusões preliminares

Reunindo os resultados obtidos através das simulações numéricas podem ser destacadas as seguintes conclusões:

- O programa desenvolvido funciona para diferentes veleiros, com base nos resultados obtidos nas simulações dos Subcapítulos 4.2 e 4.3.4;
- O programa apresenta trajetórias otimizadas de duas pernadas, em que os resultados podem ser comprovados ao estudar o diagrama polar do veleiro;





# Capítulo 5

## Conclusões e Trabalho Futuro

Neste capítulo pretende-se fazer um breve resumo do trabalho que foi realizado no desenvolvimento da dissertação de mestrado. Após o resumo vai ser feita uma análise do que foi proposto resolver e do que foi resolvido ao longo da dissertação de mestrado. Vai ser feita também uma análise dos resultados obtidos e tiradas conclusões acerca do funcionamento do código elaborado. Finalmente, seguindo-se da proposta de trabalhos que podem ser realizados futuramente.

### 5.1 Conclusões Gerais

O objetivo desta dissertação de mestrado era realizar um estudo e o desenvolvimento de um algoritmo de otimização de trajetórias para veleiros. Após o estudo dos métodos desenvolvidos anteriormente nesta área foi decidido realizar uma abordagem ao problema através da Simulação de Recozimento, que é uma técnica heurística de otimização.

Para aplicar a técnica SR o problema foi dividido em várias instâncias conforme o Subcapítulo 3.1. Neste trabalho foi abordada a Instância A1, na qual é considerado uma direção e uma força do vento constantes ao longo de todo o trajeto, para um trajetória com um máximo de duas pernadas.

A abordagem da Instância A1 foi realizada através da criação de um *script* em *MATLAB*<sup>®</sup>. De forma resumida o funcionamento do *script* é o seguinte:

1. são introduzidos os parâmetros iniciais (pontos inicial ( $A$ ) e final ( $B$ ), direção e velocidade do vento);
2. é criada uma malha base em forma de matriz que vai ser o espaço de procura;
3. são calculadas as distâncias e ângulos entre os pontos  $A$  e  $B$  e todos os restantes pontos da matriz;
4. é calculada a velocidade entre todos os pontos através dos dados do diagrama polar do veleiro e do ângulo entre esses pontos;
5. é guardado um determinado número de trajetórias que apresentem o menor tempo de trajeto;
6. é aplicado o SR que perturba os pontos e refaz os cálculos novamente;

7. é gerada a trajetória ótima de uma ou duas pernadas, consoante as condições do cenário, e os resultados são apresentados em forma gráfica.

Após as simulações numéricas realizadas em que os resultados são todos coerentes e lógicos com base no tipo de veleiro e no seu diagrama polar, pode concluir-se que a abordagem através do algoritmo desenvolvido é satisfatória e apresenta resultados matematicamente corretos.

## 5.2 Trabalhos Futuros

Como possíveis trabalhos futuros, realça-se:

- Uma melhoria no algoritmo desenvolvido para o cálculo de trajetórias ótimas com mais de duas pernadas;
- Abordagem das Instâncias A2, B, C e D com duas e mais pernadas;
- Desenvolvimento de aplicações que implementem as abordagens efetuadas nos veleiros autónomos desenvolvidos no contexto das dinâmicas “eVentos” e “Grupo de vela robótica da Escola Naval”;
- Desenvolvimento de aplicações que implementem as abordagens desenvolvidas em ambiente *Windows*, *IOS* e *Android* e sua utilização experimental nos veleiros utilizados pela Escola Naval.
- Finalmente, o desenvolvimento de outras abordagens para a resolução do mesmo problema e comparação de resultados.

# Bibliografia

- Alves, B. M. F. (2016). *Plataforma de Competição de Veleiros Autónomos (Dissertação de Mestrado)*. [https://recipp.ipp.pt/bitstream/10400.22/10606/1/DM%7B%5C\\_%7DBrunoAlves%7B%5C\\_%7D2016%7B%5C\\_%7DMEEC.pdf](https://recipp.ipp.pt/bitstream/10400.22/10606/1/DM%7B%5C_%7DBrunoAlves%7B%5C_%7D2016%7B%5C_%7DMEEC.pdf)
- Barros e Sá, N. (2005). Física Elementar da Navegação à Vela, 11. <https://www.spf.pt/magazines/GFIS/69/article/399/pdf>
- Cavaco, T. M. G. (2011). *Projecto e Ventos - Sensores (Dissertação de Mestrado)*. Alfeite.
- Cruz, N. A. & Alves, J. C. (2014). Navigation performance of an autonomous sailing robot, Em *2014 Oceans - St. John's*, Porto, IEEE. <https://doi.org/10.1109/OCEANS.2014.7003227>
- da Mota, J. M. S. (2014). *Otimização de derrotas ao campo sinóptico do vento. Aplicação prática ao N.R.P. "Sagres" (Dissertação de Mestrado)*. Alfeite. <https://comun.rcaap.pt/bitstream/10400.26/12063/1/ASPOF%20M%20Sassetti%20da%20Mota%202014.PDF>
- Engelbrecht, A. P. (2007). Optimization Theory. Em *Computational Intelligence: An Introduction* (2nd ed., pp. 551–558). John Wiley & Sons, Ltd. <https://doi.org/10.1201/b11866-5>
- Erckens, H., Beusser, G.-A., Pradalier, C. & Siegwart, R. (2010). Avalon. *IEEE Robotics & Automation Magazine*, 17(1), 45–54. <https://doi.org/10.1109/MRA.2010.935792>
- Fernandes, P. M. d. C. (2016). *Projeto e Construção de um Veleiro Autónomo, utilizando materiais compósitos, Impressão a 3D e Aprendizagem Máquina (Dissertação de Mestrado)*. Alfeite. <http://hdl.handle.net/10400.26/15068>
- Giger, L., Wismer, S., Boehl, S., Büsser, G. A., Erckens, H., Weber, J., Moser, P., Schwizer, P., Pradalier, C. & Siegwart, R. (2009). Design and Construction of the Autonomous Sailing Vessel AVALON. *Proceedings of the 2nd International Robotic Sailing Conference*, 7. <https://doi.org/10.3929/ethz-a-010035784%20Rights>
- Husmann, K., Lange, A. & Spiegel, E. (1999). *The R Package optimization : Flexible Global Optimization with Simulated-Annealing*. [https://cran.r-project.org/web/packages/optimization/vignettes/vignette%7B%5C\\_%7Dmaster.pdf](https://cran.r-project.org/web/packages/optimization/vignettes/vignette%7B%5C_%7Dmaster.pdf)
- Johnson, D. S., Aragon, R. & Mcgeoch, L. A. (1989). Optimization Annealing : an Experimental Part 1 , Graph Partitioning Evaluation. *Operations Research*, 37(6), 865–892.
- Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220(4598), 671–680. <https://doi.org/10.1126/science.220.4598.671>

- Liberti, L. & Kucherenko, S. (2005). Comparison of deterministic and stochastic approaches to global optimization. *International Transactions in Operational Research*, 12(3), 263–285. <https://doi.org/10.1111/j.1475-3995.2005.00503.x>
- Lobo, V. (2014). Envolvimento da Marinha em projetos de I&D. [https://ptdocz.com/doc/177168/6%7B%5C\\_%7Dcomunica%7B%5Cc%7Bc%7D%7D%7B%5C~%7Ba%7D%7Do%7B%5C\\_%7Dprof-sousa-%20lobo---cinav](https://ptdocz.com/doc/177168/6%7B%5C_%7Dcomunica%7B%5Cc%7Bc%7D%7D%7B%5C~%7Ba%7D%7Do%7B%5C_%7Dprof-sousa-%20lobo---cinav)
- Manley, J. E. (2008). Unmanned surface vehicles, 15 years of development, Em *OCEANS 2008*, IEEE. <https://doi.org/10.1109/OCEANS.2008.5152052>
- Manual of qtVlm version 5.9.10. (2018). [http://download.meltemus.com/qtvlm/qtVlm%7B%5C\\_%7Ddocumentation%7B%5C\\_%7Den.pdf](http://download.meltemus.com/qtvlm/qtVlm%7B%5C_%7Ddocumentation%7B%5C_%7Den.pdf)
- Martí, R. & Reinelt, G. (2011). Heuristic Methods. Em *The Linear Ordering Problem: Exact and Heuristic Methods in Combinatorial Optimization* (pp. 17–18). Berlin, Heidelberg, Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-642-16729-4\\_2](https://doi.org/10.1007/978-3-642-16729-4_2)
- MathWorks. (2005). MATLAB ® The Language of Technical Computing. Em T. Mathworks (Ed.). [https://doi.org/10.1016/S0166-6851\(96\)02621-7](https://doi.org/10.1016/S0166-6851(96)02621-7)
- Meyen, S. & Tweedie, R. (2005). *Markov Chains and Stochastic Stability* (Springer, Ed.). Verlag, Springer. <https://web.archive.org/web/20110105223013/http://probability.ca/MT/BOOK.pdf>
- Mijwil, M. (2015). Heuristic Algorithms. [https://www.researchgate.net/publication/322162995%7B%5C\\_%7DHeuristic%7B%5C\\_%7DAlgorithms](https://www.researchgate.net/publication/322162995%7B%5C_%7DHeuristic%7B%5C_%7DAlgorithms)
- OFFSHORE RACING CONGRESS (ORC) - *Speed Guide Explanation*. (2009).
- Peng, J., Huang, Y. & Luo, G. (2015). Robot Path Planning Based on Improved A\* Algorithm. *Cybernetics and Information Technologies*, 15(2), 171–180. <https://doi.org/10.1515/cait-2015-0036>
- Philpott, A. & Manson, A. (2001). Optimising Yacht Routes under Uncertainty. *The 15th Chesapeake Sailing Yacht Symposium*, 10. <http://www.inderscience.com/link.php?id=22183>
- Press, W. H., Teukolsky, S. A., Vetterling, W. T. & Flannery, B. P. (1992). Simulated annealing methods. Em *Numerical Recipes in C : The Art of Scientific Computing* (2nd Ed., pp. 444–455). USA, Cambridge University Press. [http://www.cs.ubbcluj.ro/%7B~%7Dcsatol/mestint/pdfs/Numerical%7B%5C\\_%7DRecipes%7B%5C\\_%7DSimulated%7B%5C\\_%7DAnnealing.pdf](http://www.cs.ubbcluj.ro/%7B~%7Dcsatol/mestint/pdfs/Numerical%7B%5C_%7DRecipes%7B%5C_%7DSimulated%7B%5C_%7DAnnealing.pdf)
- Rabaud, M. (2016). Optimal routing in sailing, Em *Sports Physics*, Palaiseau, France. [http://www.fast.u-psud.fr/%7B~%7Drabaud/Articles/Optimal%7B%5C\\_%7Drouting.pdf](http://www.fast.u-psud.fr/%7B~%7Drabaud/Articles/Optimal%7B%5C_%7Drouting.pdf)
- Santos, F. B. (2019). *eVentos - Desenvolvimentos no veleiro autónomo Barlavento (Dissertação de Mestrado)*. Alfeite. [https://comum.rcaap.pt/bitstream/10400.26/31236/1/BARRACOSA-eVentos%20-%20Desenvolvimentos%20no%20veleiro%20aut%7B%5C'%7Bo%7D%7Dnomo%20Barlavento%7B%5C\\_%7DV3.pdf](https://comum.rcaap.pt/bitstream/10400.26/31236/1/BARRACOSA-eVentos%20-%20Desenvolvimentos%20no%20veleiro%20aut%7B%5C'%7Bo%7D%7Dnomo%20Barlavento%7B%5C_%7DV3.pdf)
- Schmid, J. G. (1978). *Aprenda a Velejar*. [http://grayhawk.dyndns.org/vela/Dnl/Aprender%20a%20Velejar%7B%5C\\_%7Ddireito.pdf](http://grayhawk.dyndns.org/vela/Dnl/Aprender%20a%20Velejar%7B%5C_%7Ddireito.pdf)
- Stelzer, R. (2012). *Autonomous Sailboat Navigation* (tese de doutoramento). Montfort University. <http://hdl.handle.net/2086/7364>

Weise, T. (2009). *Global optimization algorithms -Theory and Application* (2nd ed.). Self-Published. <https://doi.org/10.1017/CBO9780511691881.010>



## Apêndice A - Terminologia Náutica

**Abatimento:** é o ângulo que mede o desvio do rumo do veleiro em relação à direção da sua quilha.

**Arribar:** Aproximar a proa do veleiro da linha do vento.

**Barlavento:** Lado do veleiro pelo qual entra o vento.

**Bolina cerrada:** Navegação à vela em que o vento entra no veleiro pela amura com um ângulo o mais próximo possível do ângulo de onde sopra o vento.

**Bordos:** Trajetória realizada em ziguezague alternando o bordo pelo qual entra o vento no veleiro.

**Guinar:** Alterar a proa do veleiro.

**Largo:** Tipo de navegação quando o vento entra no veleiro com um ângulo entre os noventa e os cento e setenta graus em relação à sua proa.

**Orçar:** Afastar a proa do veleiro da linha do vento.

**Patilhão:** Uma extensão da quilha com uma área transversal mínima mas com uma área longitudinal substancial.

**Pernada:** Trajeto do veleiro entre dois pontos de guinada.

**Sotavento:** Lado do veleiro pelo qual sai o vento.

**Velame:** Conjunto de todas as velas existentes no veleiro.

**Vento pela alheta:** Quando o vento entra no veleiro com um ângulo de cento e trinta e cinco graus em relação à sua proa.

**Vento pela popa:** Quando o vento entra no veleiro com um ângulo igual à sua proa.

**Vento pelo través:** Quando o vento entra no veleiro com um ângulo de noventa graus (na perpendicular) em relação à sua proa.

**Virar em roda:** Forma de mudar o bordo pelo qual entra o vento no veleiro, arribando, fazendo passar o vento pela popa do veleiro.

**Virar por d'avante:** Forma de mudar o bordo pelo qual entra o vento no veleiro, orçando, fazendo passar o vento pela proa do veleiro.

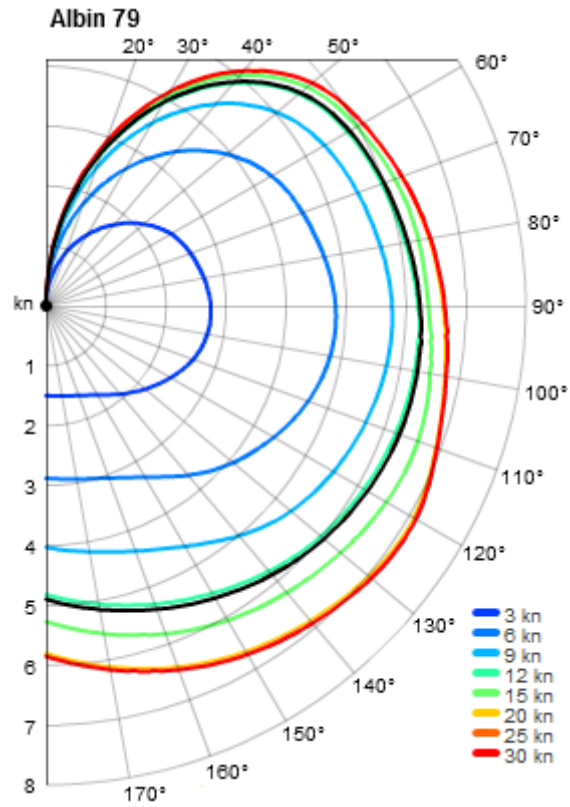




## Apêndice B - Diagramas polares de velocidade

### B.1 Diagrama polar do veleiro “*Albin 79*”

#### B.1.1 Forma Gráfica

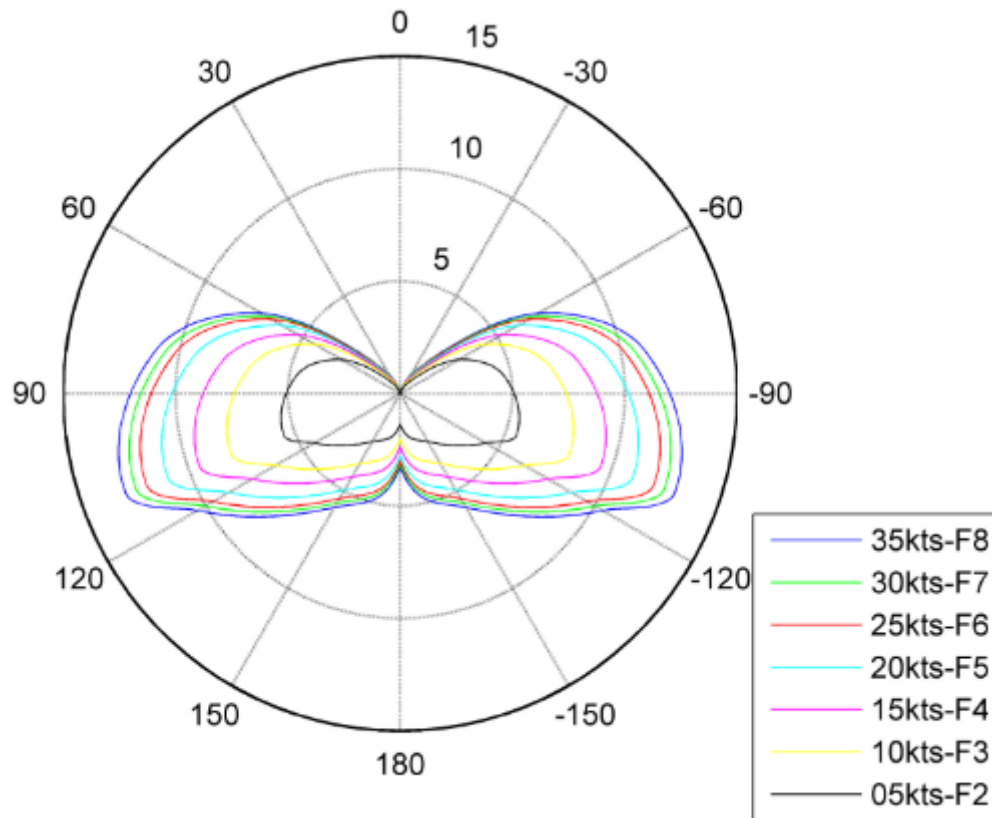


### B.1.2 Forma programada (*gera\_diag\_polar\_albin*)

```
function [diag_polar, x, y]=gera_diag_polar_albin(vento)
% Edifica o pp (piecewise polinomial) correspondente ao vento verdadeiro,
% diag_polar, que será usado nos cálculos massivos das velocidades
% Albin 79 POLAR
%
%Ângulos de entrada do vento
x=[0, 45, 60, 75, 90, 105, 120, 135, 150, 175, 180];
%Velocidades do veleiro consoante velocidade do vento
if 0 <= vento & vento < 6
    y6=[0.0,3.6,4.3,4.6,4.8,4.6,4.3,3.9,3.3,3.0,2.9];
    y=y6*vento/6;
elseif vento < 9
    y6=[0.0,3.6,4.3,4.6,4.8,4.6,4.3,3.9,3.3,3.0,2.9];
    y9=[0.0,4.8,5.4,5.6,5.8,5.6,5.5,5.2,4.6,4.2,4.0];
    y=y6+(vento-6)*(y9-y6)/3;
elseif vento < 12
    y9=[0.0,4.8,5.4,5.6,5.8,5.6,5.5,5.2,4.6,4.2,4.0];
    y12=[0.0,5.2,5.7,6.0,6.2,6.3,6.1,5.9,5.6,5.2,4.9];
    y=y9+(vento-9)*(y12-y9)/3;
elseif vento < 15
    y12=[0.0,5.1,5.7,6.0,6.2,6.3,6.1,5.9,5.6,5.2,4.9];
    y15=[0.0,5.3,6.0,6.3,6.4,6.6,6.4,6.3,6.1,5.6,5.3];
    y=y12+(vento-12)*(y15-y12)/3;
elseif vento < 20
    y15=[0.0,5.3,6.0,6.3,6.4,6.6,6.4,6.3,6.1,5.6,5.3];
    y20=[0.0,5.5,6.1,6.5,6.6,6.8,7.1,7.0,6.6,6.3,5.8];
    y=y15+(vento-15)*(y20-y15)/5;
elseif vento < 25
    y20=[0.0,5.5,6.1,6.5,6.6,6.8,7.1,7.0,6.6,6.3,5.8];
    y25=[0.0,5.5,6.1,6.5,6.6,6.8,7.2,7.0,6.6,6.3,5.9];
    y=y20+(vento-20)*(y25-y20)/5;
else
    return
end
%Interpolação final que fornece a velocidade do veleiro consoante a
%velocidade do vento
diag_polar = spline(x,y);
end
```

## B.2 Diagrama polar do N.R.P. “Sagres”

### B.2.1 Forma gráfica



## B.2.2 Forma programada (*gera\_diag\_polar\_sagres*)

```
function [diag_polar, x, y]=gera_diag_polar_sagres(vento)
% Edifica o pp (piecewise polynomial) correspondente ao vento verdadeiro,
% diag_polar, que será usado nos cálculos massivos das velocidades
% Sagres POLAR
%
%Ângulos de entrada do vento
x=[0, 30, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180];
%Velocidades do veleiro consoante velocidade do vento
if 0 <= vento & vento < 5
    y5=[0,0.2,0.8,2.5,4.5,5.0,5.8,4.2,2.5,2.3,2.0,1.7];
    y=y5*vento/5;
elseif vento < 10
    y5=[0,0.2,0.8,2.5,4.5,5.0,5.8,4.2,3.4,3.0,2.5,1.7];
    y10=[0,0.3,1.0,4.6,6.3,7.1,7.7,6.3,4.8,3.5,2.5,2.2];
    y=y5+(vento-5)*(y10-y5)/5;
elseif vento < 15
    y10=[0,0.3,1.0,4.6,6.3,7.1,7.7,6.3,4.8,3.5,3.2,2.2];
    y15=[0,0.4,1.2,5.0,7.4,8.5,9.2,7.5,6.0,4.2,3.5,2.5];
    y=y10+(vento-10)*(y15-y10)/5;
elseif vento < 20
    y15=[0,0.4,1.2,5.0,7.4,8.5,9.2,7.5,6.0,4.2,3.5,2.5];
    y20=[0,0.5,1.4,6.0,8.5,10.1,11.0,8.5,6.5,5.0,4.0,2.9];
    y=y15+(vento-15)*(y20-y15)/5;
elseif vento < 25
    y20=[0,0.5,1.4,6.0,8.5,10.1,11.0,8.5,6.5,5.0,4.0,2.9];
    y25=[0,0.6,1.6,6.4,9.5,11.0,12.1,9.6,7.3,5.4,4.5,3.2];
    y=y20+(vento-20)*(y25-y20)/5;
elseif vento < 30
    y25=[0,0.6,1.6,6.4,9.5,11.0,12.1,9.6,7.3,5.4,4.5,3.2];
    y30=[0,0.7,1.7,6.6,10.0,11.4,12.5,10.0,7.4,5.6,4.8,3.3];
    y=y25+(vento-25)*(y30-y25)/5;
elseif vento < 35
    y30=[0,0.7,1.7,6.6,10.0,11.4,12.5,10.0,7.4,5.8,4.8,3.3];
    y35=[0,0.8,1.8,7.2,10.4,12.1,12.7,10.4,7.5,6.0,5.0,3.5];
    y=y30+(vento-30)*(y35-y30)/5;
else
    return
end
%Interpolação final que fornece a velocidade do veleiro consoante a
%velocidade do vento
diag_polar = spline(x,y);
end
```

## B.3 Diagrama polar do veleiro *Alba*

### B.3.1 Forma programada (*gera\_diag\_polar\_alba*)

```
function [diag_polar, x, y]=gera_diag_polar(vento)
% Edifica o pp (piecewise polinomial) correspondente ao vento verdadeiro,
% diag_polar, que será usado nos cálculos massivos das velocidades
% ALBA POLAR
%
%Ângulos de entrada do vento
x=[0, 45, 60, 75, 90, 105, 120, 135, 150, 175, 180];
%Velocidades do veleiro consoante velocidade do vento
if 0 <= vento & vento < 6
    y6=[0.0,1.0,1.2,1.6,2.0,1.8,1.6,1.4,1.3,1.2,1.0];
    y=y6*vento/6;
elseif vento < 8
    y6=[0.0,1.0,1.2,1.6,2.0,1.8,1.6,1.4,1.3,1.2,1.0];
    y8=[0.0,2.0,3.0,3.2,3.2,3.0,2.8,2.6,2.4,2.2,2.2];
    y=y6+(vento-6)*(y8-y6)/2;
elseif vento < 10
    y8=[0.0,2.0,3.0,3.2,3.2,3.0,2.8,2.6,2.4,2.2,2.2];
    y10=[0.0,3.0,4.0,4.7,5.0,4.8,4.0,3.8,3.6,3.2,3.0];
    y=y8+(vento-8)*(y10-y8)/2;
elseif vento < 12
    y10=[0.0,3.0,4.0,4.7,5.0,4.8,4.0,3.8,3.6,3.2,3.0];
    y12=[0.0,3.2,4.5,5.4,5.5,5.4,5.2,4.8,4.4,4.2,4.0];
    y=y10+(vento-10)*(y12-y10)/2;
elseif vento < 14
    y12=[0.0,3.2,4.5,5.4,5.5,5.4,5.2,4.8,4.4,4.2,4.0];
    y14=[0.0,3.5,5.0,6.0,6.0,5.6,5.4,5.0,4.8,4.7,4.5];
    y=y12+(vento-12)*(y14-y12)/2;
elseif vento < 16
    y14=[0.0,3.5,5.0,6.0,6.0,5.6,5.4,5.0,4.8,4.7,4.5];
    y16=[0.0,3.6,5.3,6.3,6.5,6.3,6.0,5.8,5.5,5.3,5.0];
    y=y14+(vento-14)*(y16-y14)/2;
elseif vento < 20
    y16=[0.0,3.6,5.3,6.3,6.5,6.3,6.0,5.8,5.5,5.3,5.0];
    y20=[0.0,4.0,5.5,6.5,7.0,6.8,6.5,6.2,5.8,5.6,5.5];
    y=y16+(vento-16)*(y20-y16)/4;
elseif vento <= 25
    y20=[0.0,4.0,5.5,6.5,7.0,6.8,6.5,6.2,5.8,5.6,5.5];
    y25=[0.0,4.2,5.8,6.8,7.2,7.2,7.0,6.7,6.4,6.2,6.0];
    y=y20+(vento-20)*(y25-y20)/5;
else
    return
end
%Interpolação final que fornece a velocidade do veleiro consoante a
%velocidade do vento
diag_polar = spline(x,y);
end
```

## B.4 Diagrama polar dos veleiro tipo *SB 20*

### B.4.1 Forma programada (*gera\_diag\_polar\_sb20*)

```
function [diag_polar, x, y]=gera_diag_polar_sb20(vento)
% Edifica o pp (piecewise polinomial) correspondente ao vento verdadeiro,
% diag_polar, que será usado nos cálculos massivos das velocidades
% SB 20 POLAR
%
%Ângulos de entrada do vento
x=[0, 52, 60, 75, 90, 110, 120, 135, 150, 180];
%Velocidades do veleiro consoante velocidade do vento
if 0 <= vento & vento < 6
    y6=[0.0,4.5,4.78,5.02,5.29,5.33,5.20,4.72,4.04,3.49];
    y=y6*vento/6;
elseif vento < 8
    y6=[0.0,4.5,4.78,5.02,5.29,5.33,5.20,4.72,4.04,3.49];
    y8=[0.0,5.27,5.46,5.61,5.88,5.95,5.85,5.57,5.01,5.01];
    y=y6+(vento-6)*(y8-y6)/2;
elseif vento < 10
    y8=[0.0,5.27,5.46,5.61,5.88,5.95,5.85,5.57,5.01,5.01];
    y10=[0.0,5.63,5.8,5.97,6.17,6.59,6.47,6.10,5.6,4.85];
    y=y8+(vento-8)*(y10-y8)/2;
elseif vento < 12
    y10=[0.0,5.63,5.8,5.97,6.17,6.59,6.47,6.10,5.6,4.85];
    y12=[0.0,5.82,6.07,6.39,6.45,7.01,7.23,6.71,5.99,5.29];
    y=y10+(vento-10)*(y12-y10)/2;
elseif vento < 14
    y12=[0.0,5.82,6.07,6.39,6.45,7.01,7.23,6.71,5.99,5.29];
    y14=[0.0,5.9,6.2,6.73,6.94,7.38,7.81,7.47,6.44,5.72];
    y=y12+(vento-12)*(y14-y12)/2;
elseif vento < 16
    y14=[0.0,5.9,6.2,6.73,6.94,7.38,7.81,7.47,6.44,5.72];
    y16=[0.0,5.95,6.27,6.89,7.47,7.8,8.3,8.47,7.16,6.2];
    y=y14+(vento-14)*(y16-y14)/2;
elseif vento < 20
    y16=[0.0,5.95,6.27,6.89,7.47,7.8,8.3,8.47,7.16,6.2];
    y20=[0.0,6.07,6.44,7.03,7.92,8.83,9.81,11.91,10.15,8.79];
    y=y16+(vento-16)*(y20-y16)/4;
else
    return
end
%Interpolação final que fornece a velocidade do veleiro consoante a
%velocidade do vento
diag_polar = spline(x,y);
end
```

## Apêndice C - Programa principal e subprogramas

### C.1 Programa principal (*main\_mf.m*)

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% main_mf
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc; close all; clear all;
%
iprint=0; % FAZER iprint=1 para gravar os gráficos produzidos
%
x1=0; y1=0; % Coordenadas do Ponto A
x2=100; y2=0; % Coordenadas do ponto B
%
A=[x1 y1]; % PONTO A
B=[x2 y2]; % PONTO B
%
%Velocidade do vento em nós
vento=13;
% Imposição do valor do Azimute Para Onde Sopra o Vento ("aposv") que
% permitirá definir o ângulo theta correspondente ao rumo de cada pernada,
% (rAP e rPB) para entrada no diagrama polar das velocidades do veleiro.
aposv=270; %Azimute para onde sopra o vento
A=[x1 y1]; % PONTO A
B=[x2 y2]; % PONTO B
%
% A variável "passos" definirá uma distância de referência "resolucao"
% para gerar a malha base (distância entre "linhas" e "colunas") e para
% a "perturbação" de representantes da poluplação de trajectos possíveis.
% A distância de referência = dAB/passos;
passos=5; % número de subdivisões de dAB (distância de referência)
%
%
% A função malhabase estabelece a "grid" inicial
[Xbase, Ybase, dAB, rAB, resolucao]=malhabase(A,B,passos);
%
%
% A função trajecto1P gera trajectos com 1 viragem (2 pernadas)
% Cada ponto da "grid" define um ponto de viragem
% Calcula as distância dAP, dPB e os azimutes das pernadas rAP e rPB
% em coordenadas polares [-180 a +180]
[dAP,rAP, dPB, rPB]=trajecto1P(Xbase,Ybase,A,B);
%
```

```
% Calculo dos correspondentes ângulos theta de entrada no diagrama polar
% para computação das velocidades verificadas nas pernadas e de A para B.
[thetaAB, adosv]=theta0P(rAB,aposv);
[thetaAP,thetaPB]=theta1P(rAP,rPB,aposv);
%
%
%
% Gera_diag_polar com base nos dados deste tipo de diagrama fornecido pelo
% veleiro. Este diagrama polar está associado à velocidade do vento "vento"
% e será utilizado em todos os cálculos feitos daqui para a frente.
% diag_polar é uma função polinomial (pp) definida por ramos de 0 a 180°.
[diag_polar,x,y]=gera_diag_polar_sagres(vento)
%
%
% Cálculo das velocidades em cada pernada e da navegação direta
%Velocidade do ponto A ao ponto P
vel_AP=ppval(diag_polar,thetaAP);
%Velocidade do ponto B ao ponto P
vel_PB=ppval(diag_polar,thetaPB);
%Velocidade do ponto A ao ponto B
vel_AB=ppval(diag_polar,thetaAB)
%
%
%Cálculo dos tempos de percurso em cada pernada
%Tempo de navegação do ponto A ao ponto P
temp_AP=dAP./vel_AP;
%Tempo de navegação do ponto P ao ponto B
temp_PB=dPB./vel_PB;
%
%
% Calculo dos tempos totais de percurso (horas) (distância em milhas
% nauticas e velocidade em nós)
%
%Tempo total navegação com duas pernadas
tempo_total_APB=temp_AP+temp_PB
%Tempo total navegação direta
tempo_total_AB=dAB/vel_AB;
%
%
%
```



```

%% INICIO SIMULATING ANNEALING
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Colocar os valores do tempo total por ordem crescente através da função
%"sort" e transformar a matriz num vetor.
%Xbase, Ybase, tempo_total_APB
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[tempo_total_APB_C id]=sort(tempo_total_APB(:));
%
%
%Colocar os valores das coordenadas de acordo com os valores do tempo,
%transformando a matriz num vetor
Xbase_C=Xbase(id);
Ybase_C=Ybase(id);
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Dimensão("val") do espaço a minimizar 4sqrt(m*n)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
[l,c]=size(tempo_total_APB);
%
val=round(4*sqrt(l*c));
%
%Guardar "val" resultados
tempo_total_APB_1=tempo_total_APB_C(1:val);
Xvector_1=Xbase_C(1:val);
Yvector_1=Ybase_C(1:val);
%
%
%Guardar o melhor resultado
melhor_tempo=tempo_total_APB_1(1);
melhor_X=Xvector_1(1);
melhor_Y=Yvector_1(1);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Amplitude perturbadora inicial amp_b=resolucao/2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
amp_b=resolucao/2;
%
tempo_min=[];
%
epocas=100; % numero de iterações

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Início do loop
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for il=1:epocas
%Guardar o melhor tempo da geração anterior e as correspondentes
%coordenadas
total_tempo_tmp=melhor_tempo;
Xvector_tmp=melhor_X;
Yvector_tmp=melhor_Y;
%
%Diminuição da amplitude perturbadora ao longo das iterações
amp_b=amp_b/1.01;
%
n_min=-amp_b;
n_max=amp_b;
%
%
%Criação dos vetores que vão fazer a perturbação
Xrandom=n_min+(n_max-n_min)*rand(val,1); %amplitude corrente X
Yrandom=n_min+(n_max-n_min)*rand(val,1); %amplitude corrente Y
%Realização da perturbação nos resultados existentes
%
Xvector_1=plus(Xvector_1,Xrandom);
Yvector_1=plus(Yvector_1,Yrandom);
%
%Calcular os novos resultados
%
[dAP_1,rAP_1, dPB_1, rPB_1]=trajecto1P(Xvector_1,Yvector_1,A,B);
[thetaAP_1,thetaPB_1]=theta1P(rAP_1,rPB_1,aposv);
vel_AP_1=ppval(diag_polar,thetaAP_1);
vel_PB_1=ppval(diag_polar,thetaPB_1);
temp_AP_1=dAP_1./vel_AP_1;
temp_PB_1=dPB_1./vel_PB_1;
tempo_total_APB_1=temp_AP_1+temp_PB_1;
% Preservar o melhor representante anterior
[tempo_total_APB_1 id]=sort(tempo_total_APB_1);
Xvector_1=Xvector_1(id);
Yvector_1=Yvector_1(id);
%

```

```

%Substituir o ultimo valor desta geração pelo primeiro da geração
%anterior e fazer o "sort"
tempo_total_APB_1(end)=total_tempo_tmp;
Xvector_1(end)=Xvector_tmp;
Yvector_1(end)=Yvector_tmp;
%
[tempo_total_APB_1 id]=sort(tempo_total_APB_1);
Xvector_1=Xvector_1(id);
Yvector_1=Yvector_1(id);
%
%
%Guardar o melhor resultado desta geração para ser usado na geração
%seguinte
melhor_tempo=tempo_total_APB_1(1);
melhor_X=Xvector_1(1);
melhor_Y=Yvector_1(1);
%
tempo_min=[tempo_min, melhor_tempo]; %Criado para ver a evolução do melhor
%tempo de geração para geração
end
%
%
%Apresentação do tempo de trajeto A-B e A-P-B ótimo em hh:mm
tempo_total_AB_h=duration(hours(tempo_total_AB));
tempo_total_AB_h=string(tempo_total_AB_h, 'hh:mm')
%
%
%Cálculo da velocidade média dos trajetos finais
[dap, rap, dpb, rpb]=trajectolP(melhor_X, melhor_Y, A, B);
dapb=dap+dpb;
dapb=round(dapb,1)
vel_med_apb=dapb/melhor_tempo;
vel_med_apb=round(vel_med_apb,1)
%
%
```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Apresentação dos resultados em forma de gráficos
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Plot para ver a evolução do tempo mínimo
figure(1)
plot(tempo_min, '-k', 'color', 'red', 'LineWidth', 1.5);
set(gca, 'FontSize', 10, 'FontWeight', 'bold', 'LineWidth', 2.0, 'box', 'on') ;
a1=xlabel(['\it Ciclos ', '\rm [n°]']); set(a1, 'FontSize', 12, 'FontWeight'...
    , 'bold');
a2=ylabel(['\it T', '\rm [h]']); set(a2, 'FontSize', 12, 'FontWeight', 'bold');
a3=title('Evolução do tempo mínimo com o decorrer dos ciclos');
set(a3, 'FontSize', 12, 'FontWeight', 'bold')
grid on
%
%Guardar o gráficos
if iprint==1;
ref_h=clock;
fName1=['ano_', num2str(ref_h(1)), '_mes_', num2str(ref_h(2)), '_dia_', ...
    num2str(ref_h(3)), '_hora_', num2str(ref_h(4)), '_min_', ...
    num2str(ref_h(5))];
fig1 = ['EVOL_CAL_FIG_PS_', num2str(fName1)]; print(gcf, '-dpsc', fig1)
fig2 = ['EVOL_CAL_FIG_DMETA_', num2str(fName1)]; print(gcf, '-dmeta', fig2)
fig3 = ['EVOL_CAL_FIG_EPS_', num2str(fName1)]; print(gcf, '-depsc', fig3)
fig4 = ['EVOL_CAL_FIG_MAT_', num2str(fName1)]; savefig(fig4)
%
end
%
%Plot dos trajectos A-B e A-P-B
figure(2)
pontos=plot(A(1), A(2), 'x', melhor_X(1), melhor_Y(1), 'o', B(1), B(2), 's');
set(pontos, 'MarkerSize', 8, 'MarkerEdgeColor', 'k', 'LineWidth', 1.5);
set(gca, 'FontSize', 10, 'FontWeight', 'bold', 'LineWidth', 2.0, 'box', 'on') ;
b1=xlabel('Coordenada X'); set(b1, 'FontSize', 12, 'FontWeight', 'bold')
b2=ylabel('Coordenada Y'); set(b2, 'FontSize', 12, 'FontWeight', 'bold')
b3=title('Trajetos A-B e A-P-B'); set(b3, 'FontSize', 12, 'FontWeight', 'bold')
grid on
hold on
quiver1=quiver(A(1), A(2), B(1)-A(1), B(2)-A(2), 0);
set(quiver1, 'MarkerSize', 8, 'MarkerEdgeColor', 'k', 'LineWidth', 1.5);
hold on
plot([A(1) melhor_X(1)], [A(2) melhor_Y(1)], '--k', 'LineWidth', 1.5)
quiver(A(1), A(2), melhor_X(1)-A(1), melhor_Y(1)-A(2), 0)
hold on
plot([melhor_X(1) B(1)], [melhor_Y(1) B(2)], '--k', 'LineWidth', 1.5)
quiver(melhor_X(1), melhor_Y(1), B(1)-melhor_X(1), B(2)-melhor_Y(1), 0)
axis equal
hold on

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   VENTO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
px=sind(aposv); py=cosd(aposv);
quiver(Xbase,Ybase,0*Xbase+px,0*Ybase+py,'r')
legend('A','P','B')
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Legendas
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
z1=text(Xbase(15,15),Ybase(15,15)-resolucao/1.9,...
        ['Vel do vento: ', num2str(vento), ' nós',]);set(z1,'Color','red',...
        'FontSize',9,'FontWeight','bold');
%
z2=text(A(1),A(2)+resolucao/4,['A: (',num2str(x1),',',...
        num2str(y1),')']);set(z2,'FontSize',9,'FontWeight','bold')
%
melhor_X=roundn(melhor_X,0);
melhor_Y=roundn(melhor_Y,0);
z3=text(melhor_X(1),melhor_Y(1)-resolucao/4,['P: (',num2str(melhor_X)...
        ', ',num2str(melhor_Y),')']);set(z3,'FontSize',9,'FontWeight','bold')
%
z4=text(B(1),B(2)-resolucao/4,['B: (',num2str(x2),',',...
        num2str(y2),')']);set(z4,'FontSize',9,'FontWeight','bold')
%
%
%
%
%Guardar o gráfico
if iprint==1;
fName2=['ano_',num2str(ref_h(1)),'_mes_',num2str(ref_h(2)),'_dia_',...
        num2str(ref_h(3)),'_hora_',num2str(ref_h(4)),'_min_',...
        num2str(ref_h(5))];
fig5 = ['TRAJ_OPTIM_FIG_PS_',num2str(fName2)]; print(gcf, '-dpsc', fig5 )
fig6 = ['TRAJ_OPTIM_FIG_DMETA_',num2str(fName2)]; print(gcf, '-dmeta',...
        fig6)
fig7 = ['TRAJ_OPTIM_FIG_EPS_',num2str(fName2)]; print(gcf, '-depsc', fig7)
fig8 = ['TRAJ_OPTIM_FIG_MAT_',num2str(fName2)]; savefig(fig8)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end

```

## C.2 Subprogramas

### C.2.1 Malha base (*malhabase.m*)

```
function [Xbase, Ybase, dAB, rAB, resolucao]=malhabase(A,B,passos)
% dAB-distância de A a B
% rAB-rumo de A a B (coordenadas polares [-180 a +180])
dAB=pdist2(A,B,'euclidean');
rAB=atan2d(B(2)-A(2),B(1)-A(1));
%Resolução que devine a distância entre os pontos na malha
resolucao=dAB/passos;
%Eixos da malha
menorX=min(A(1),B(1));
maiorX=max(A(1),B(1));
menorY=min(A(2),B(2));
maiorY=max(A(2),B(2));
%Criação da malha
vectorX=menorX-dAB:resolucao:maiorX+dAB;
vectorY=menorY-dAB:resolucao:maiorX+dAB;
%
[Xbase,Ybase] = meshgrid(vectorX,vectorY);
end
```

### C.2.2 Rumo A-B (*theta0P.m*)

```
function [thetaAB, adosv]=theta0P(rAB,aposv)
%rAB rumo de A para B em coordenadas polares [-180 a +180]
%aposv-Azimute Para Onde Sopra o Vento
%adosv-Azimute De Onde Sopra o Vento
% thetaAB Ângulos de entrada no diagrama polar

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Conversão de "aposv" (0[N] a 360) em "adosv" [-180 0[E] +180[
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

aposv_cp=atan2d(cosd(aposv),sind(aposv));%Az aposv em coordenadas polares
if aposv_cp >= 0
    adosv=aposv_cp-180;
else
    adosv=aposv_cp+180
end

thetaAB=acosd(cosd(rAB)*cosd(adosv)+sind(rAB)*sind(adosv));
end
```

### C.2.3 Trajeto A-P-B (*trajeto1P.m*)

```
function [dAP,rAP, dPB, rPB]=trajeto1P(Xbase,Ybase,A,B)
%rAB e rPB rumos das pernadas relativamente a LESTE
%
[m,n] = size(Xbase); % Determina as dimensões da matriz Xbase
Amx=ones(m,n)*A(1); % Matriz cujas entradas são a 1ª coordenada de A
Amy=ones(m,n)*A(2); % Matriz cujas entradas são a 2ª coordenada de A
Bmx=ones(m,n)*B(1); % Matriz cujas entradas são a 1ª coordenada de B
Bmy=ones(m,n)*B(2); % Matriz cujas entradas são a 1ª coordenada de B
%
% Calcular as distâncias do ponto A ao ponto P
dAP=sqrt((Xbase-Amx).^2+(Ybase-Amy).^2);
% Calcular as distâncias do ponto P ao ponto B
dPB=sqrt((Xbase-Bmx).^2+(Ybase-Bmy).^2);
%
%Calcular o rumo do ponto A ao ponto P
rAP=atan2d(Ybase-Amy,Xbase-Amx);
%Calcular o rumo do ponto A ao ponto P
rPB=atan2d(Bmy-Ybase,Bmx-Xbase);
end
```

### C.2.4 Rumo $A-P$ e $P-B$ ( $theta1P.m$ )

```
function [thetaAP,thetaPB]=theta1P(rAP,rPB,aposv)
%rAP e rPB rumos em coordenadas polares [-180 a +180]
%aposv-Azimute Para Onde Sopra o Vento
%adosv-Azimute De Onde Sopra o Vento
% thetaAP,thetaPB Ângulos de entrada nos diagramas polares (das pernadas)

[m,n] = size(rAP); % Determina as dimensões da matriz rAP

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Conversão de "aposv" (0[N] a 360) em "adosv" [-180 0[E] +180[
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

aposv_cp=atan2d(cosd(aposv),sind(aposv));%Az aposv em coordenadas polares
if aposv_cp >= 0
    adosv=aposv_cp-180;
else
    adosv=aposv_cp+180;
end

adosv=ones(m,n)*adosv;

thetaAP=acosd(cosd(rAP).*cosd(adosv)+sind(rAP).*sind(adosv));
thetaPB=acosd(cosd(rPB).*cosd(adosv)+sind(rPB).*sind(adosv));

end
```

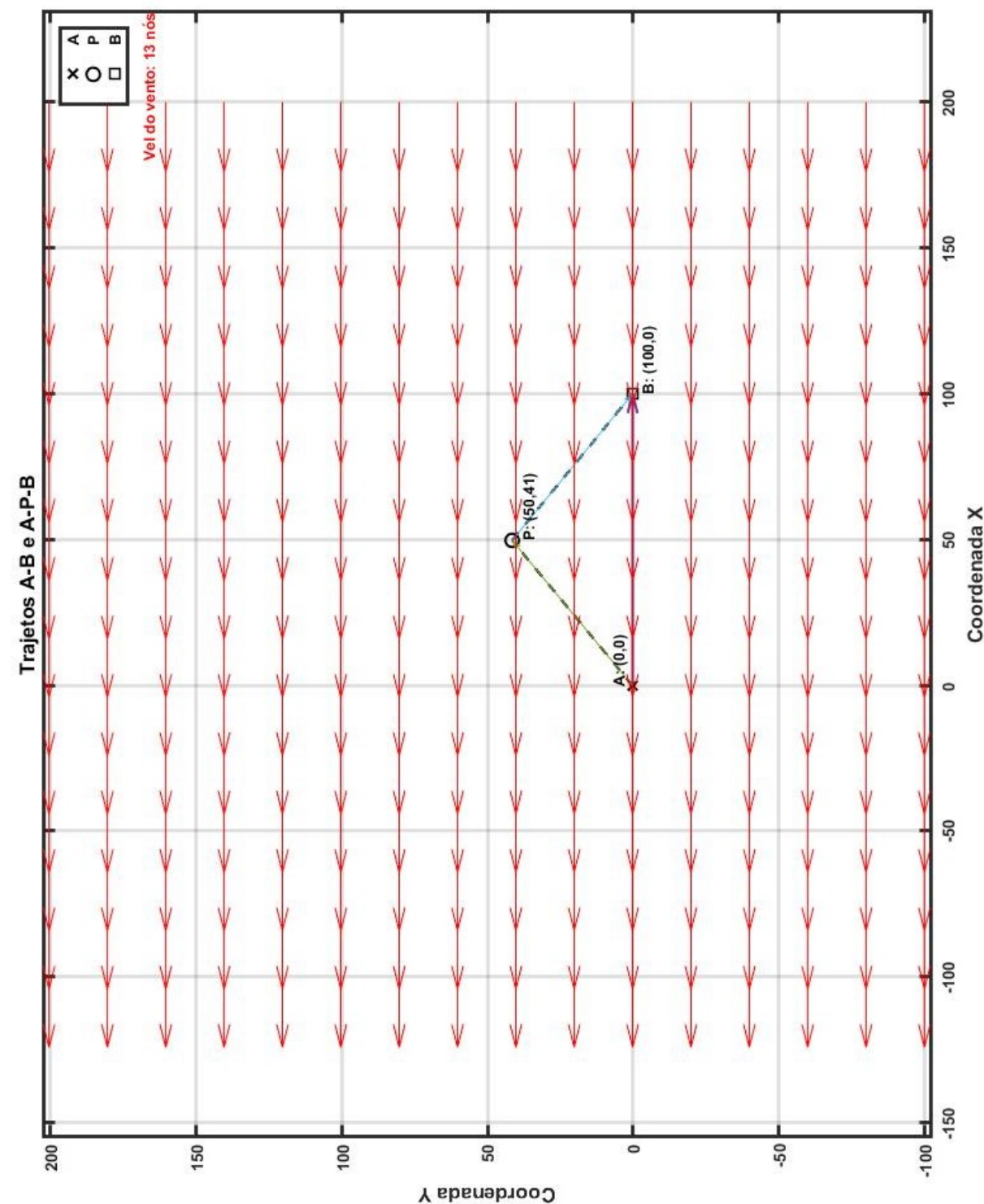
### C.2.5 Geração do diagrama polar ( $gera\_diag\_polar$ )

Apresentados no Apêndice B.

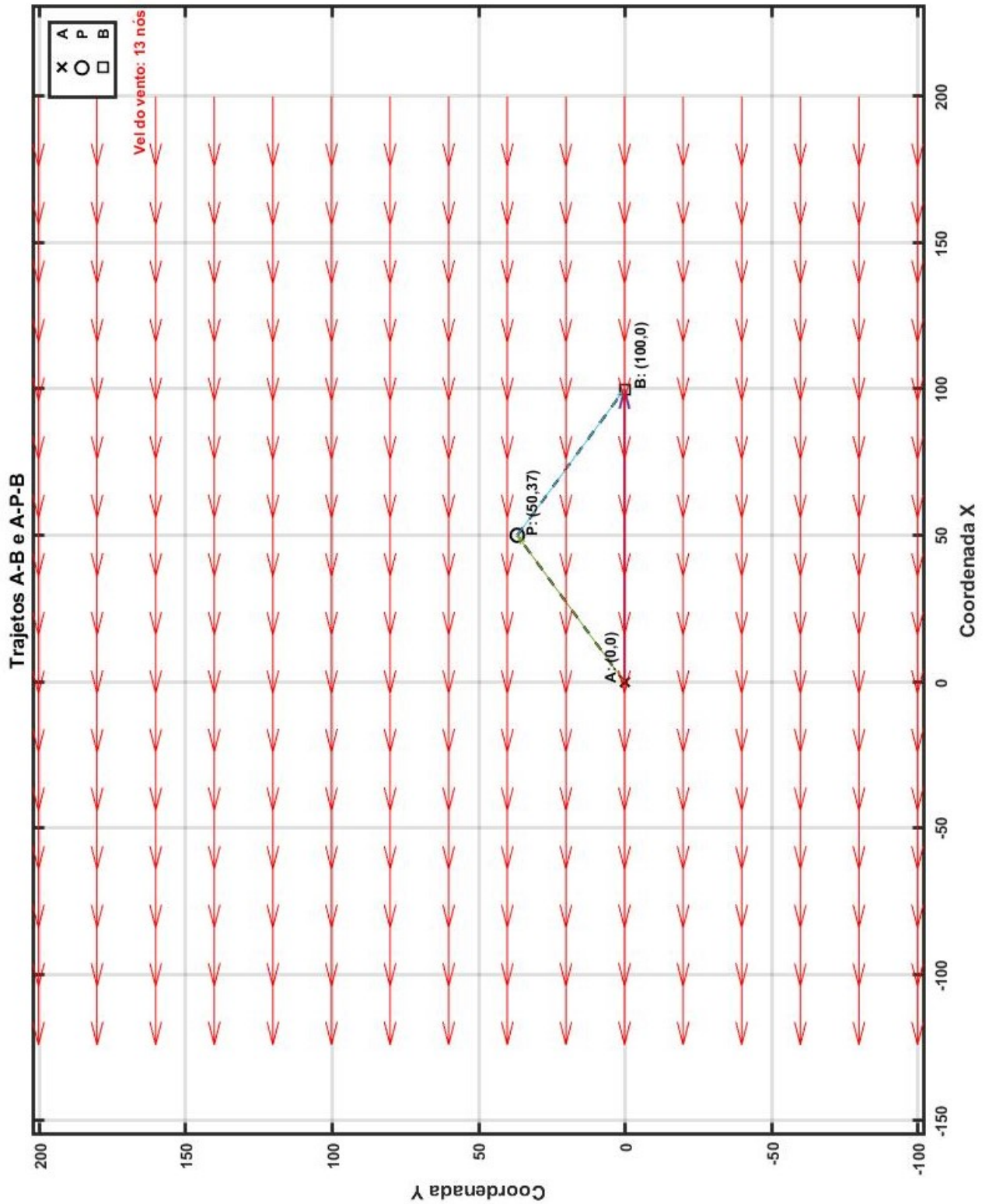


Apêndice D - Gráficos

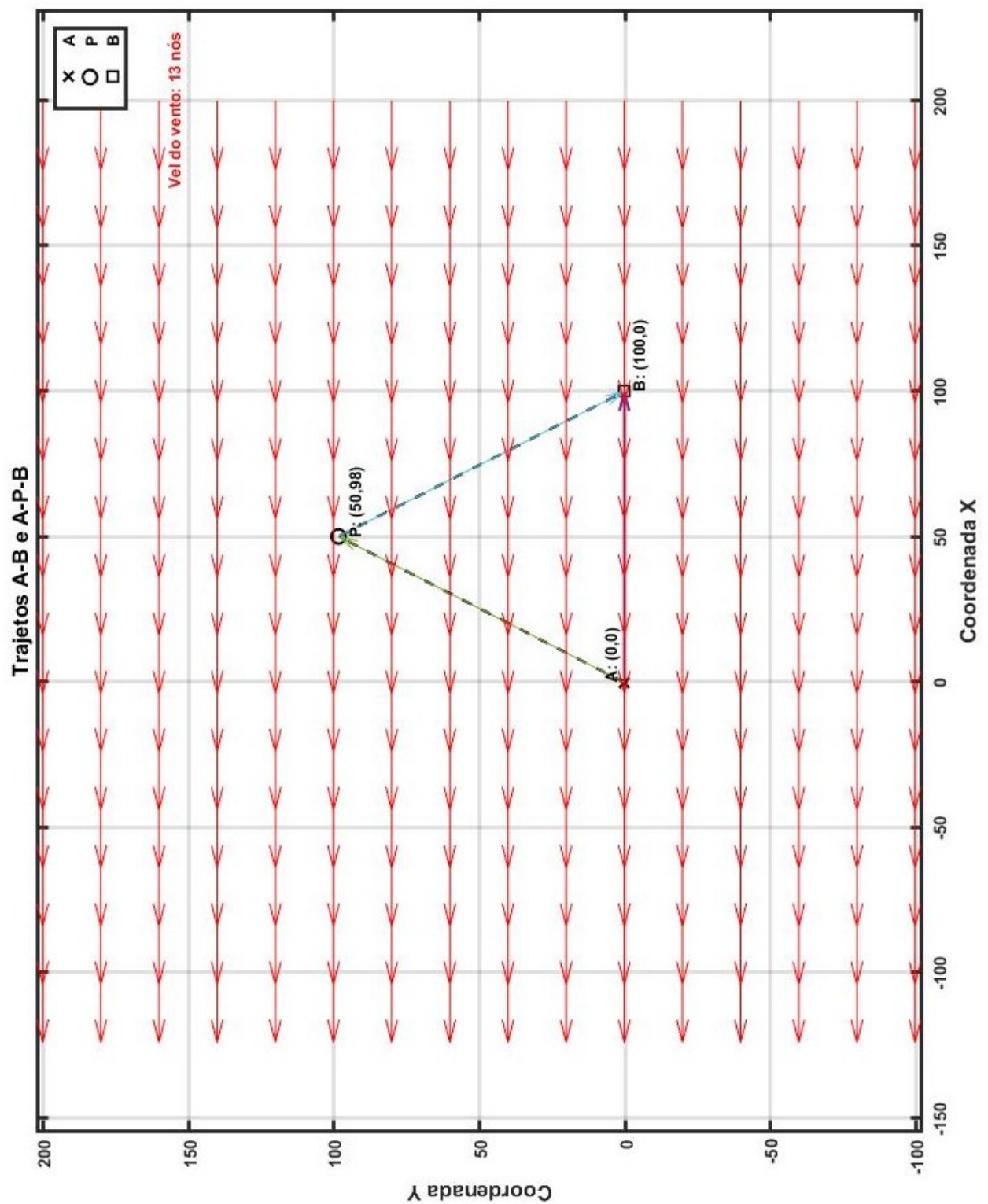
D.1 Navegação contra o vento do veleiro do tipo “Albin 79”



## D.2 Navegação contra o vento do veleiro do tipo “SB 20”

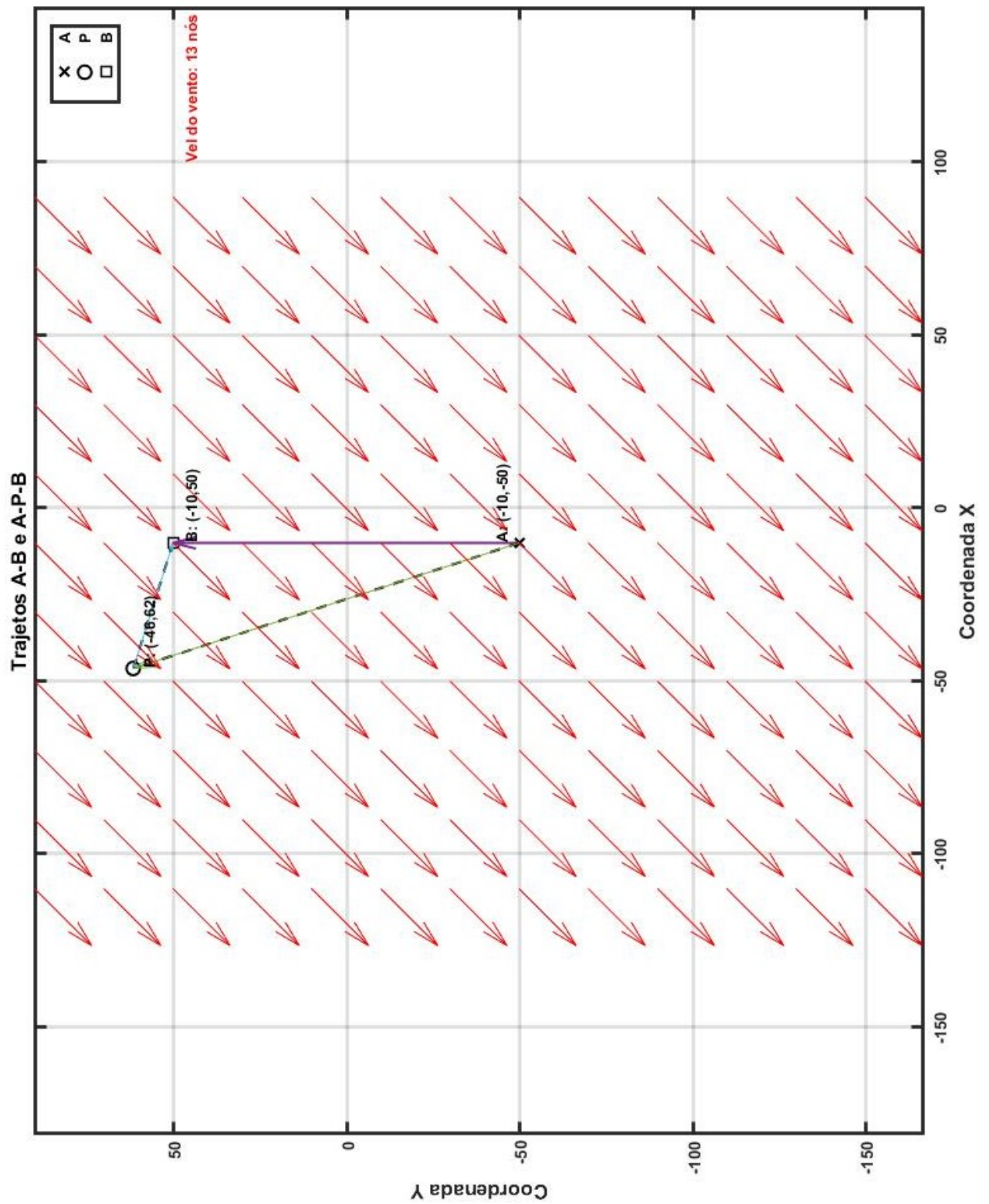


### D.3 Navegação contra o vento do N.R.P. “Sagres”

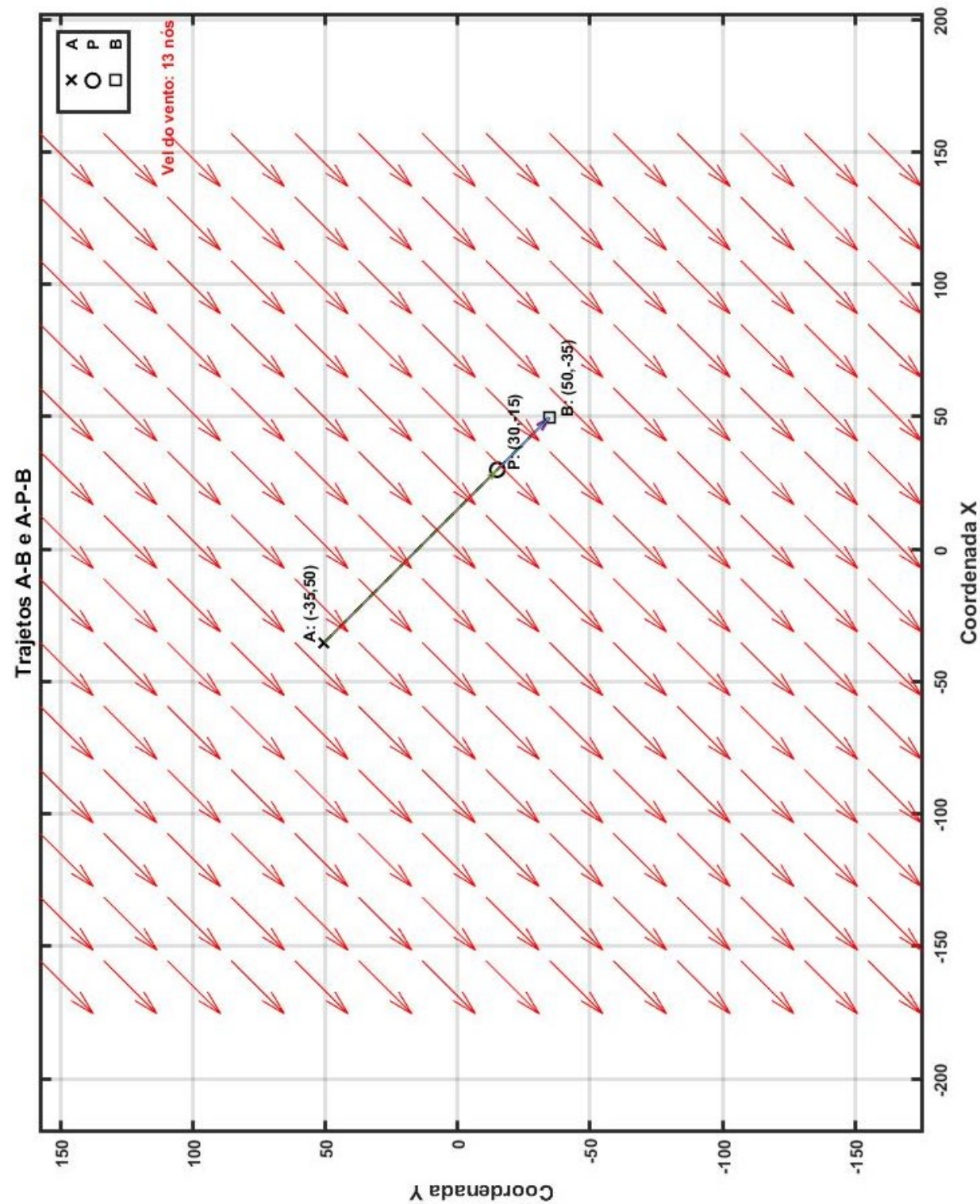




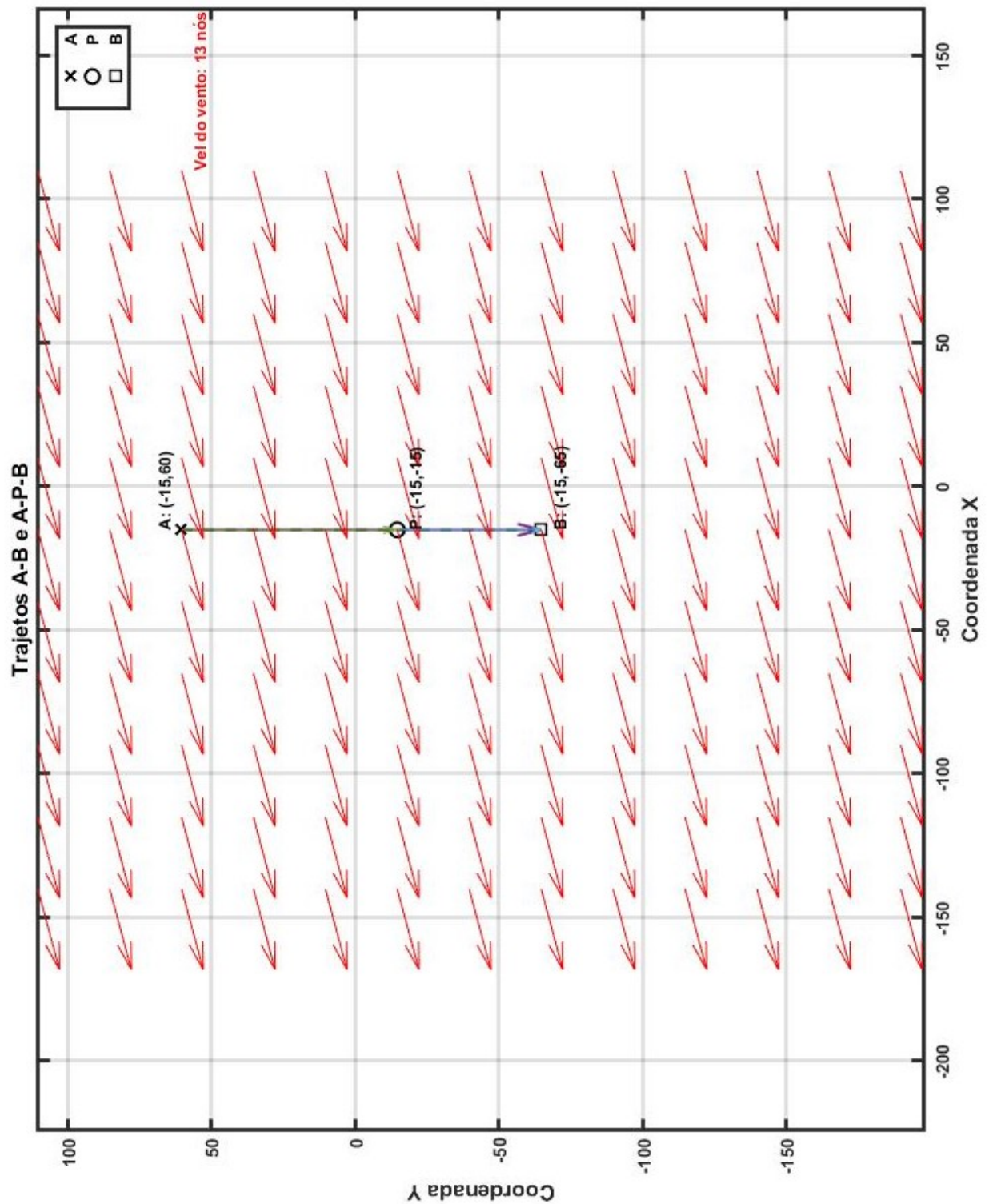
## D.4 Navegação à bolina cerrada do N.R.P. "Sagres"



D.5 Navegação com o vento pelo través do veleiro N.R.P. "Sagres "

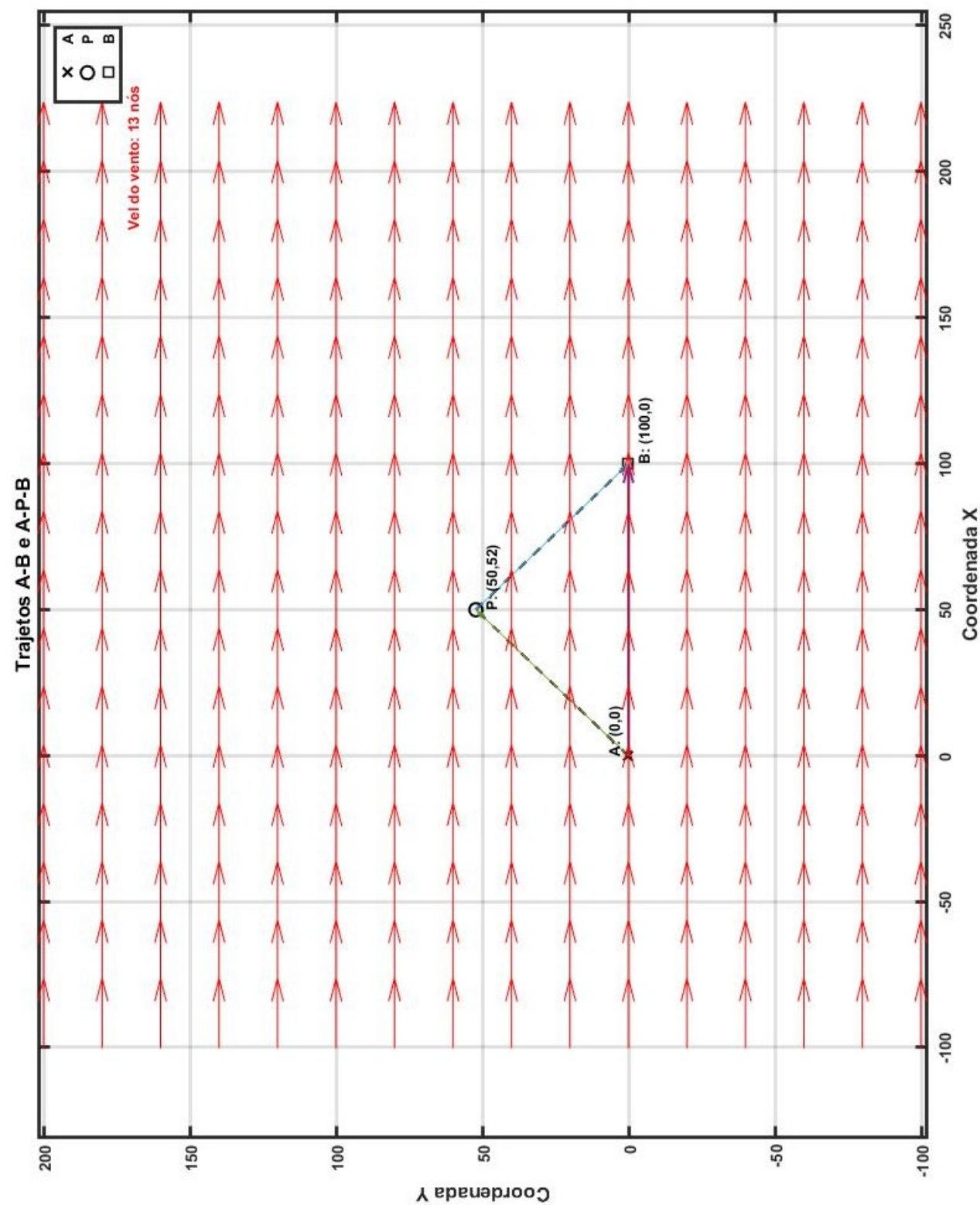


## D.6 Navegação com o vento na alheta do veleiro N.R.P. "Sagres "






D.7 Navegação à popa do veleiro N.R.P. "Sagres"








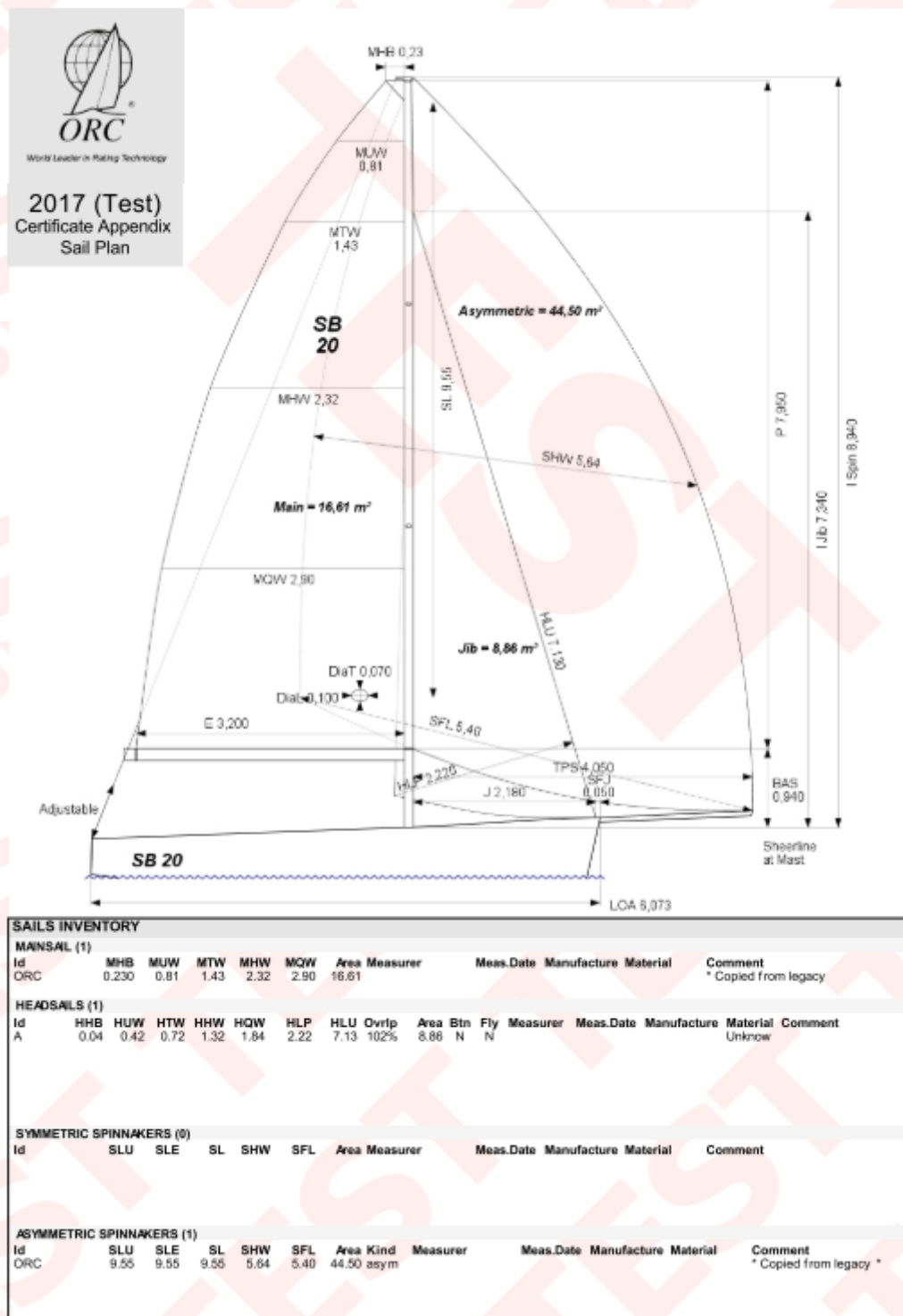
# Anexo I - Ficha técnica do veleiro "SB 20"

<b>BOAT</b> Name <b>SB 20</b> Sail Nr <b>SB20-</b>		<b>GPH</b> <b>716.9</b>	<b>HULL</b> Length Overall <b>6.073m</b> Maximum Beam <b>2.064m</b> Displacement <b>695kg</b> Draft <b>1.504m</b> IMS Reg. Division <b>Sportboat</b> Dynamic Allowance <b>0.038%</b> Fwd Accommodation <b>No</b> Hull Construction <b>Solid</b> Carbon Rudder <b>No</b> Crew Arm Extension <b>0.50</b> IMSL <b>5.913m</b> VCGD <b>-0.136m</b> Sink <b>6.42kg/mm</b> RL <b>5.829m</b> VCGM <b>-0.144m</b> WS <b>8.37m²</b> LSMO <b>5.747m</b> Displacement/Length ratio <b>3.6615</b>		 World Leader in Rating Technology <b>2017 (Test)</b> ORC International One Design Certificate		
<b>GENERAL</b> Class <b>SB20 ONE DESIGN</b> Designer <b>TONY CASTRO</b> Builder <b>LASER</b> Series <b>11/2001</b> Age <b>01/2006</b> Age Allowance <b>0.487%</b> Offset File <b>LASB3.OD - 19/01/2017 17:09:00</b> Measurement by <b>MAUGERI - 16/03/2012</b>							
<b>SCORING OPTIONS</b>							
Time On Distance	<b>699.5</b>			<b>794.5</b>			
Time On Time	<b>0.8577</b>			<b>0.8496</b>			
Triple Number	Low	Medium	High	Low	Medium	High	
Time on Distance	<b>801.1</b>	<b>645.1</b>	<b>559.9</b>	<b>1026.3</b>	<b>799.1</b>	<b>693.8</b>	
Time on Time	<b>0.8426</b>	<b>1.0464</b>	<b>1.2056</b>	<b>0.6577</b>	<b>0.8447</b>	<b>0.9729</b>	
<b>TIME ALLOWANCES</b>							
Wind Velocity	6 kt	8 kt	10 kt	12 kt	14 kt	16 kt	20 kt
Beat VMG	<b>1223.7</b>	<b>1022.1</b>	<b>916.2</b>	<b>874.6</b>	<b>863.1</b>	<b>859.8</b>	<b>855.8</b>
52°	<b>800.8</b>	<b>683.3</b>	<b>639.0</b>	<b>618.1</b>	<b>609.7</b>	<b>605.4</b>	<b>592.9</b>
60°	<b>753.9</b>	<b>659.4</b>	<b>621.2</b>	<b>593.1</b>	<b>580.7</b>	<b>573.9</b>	<b>559.4</b>
75°	<b>716.8</b>	<b>641.7</b>	<b>603.0</b>	<b>563.0</b>	<b>534.7</b>	<b>522.5</b>	<b>512.0</b>
90°	<b>680.1</b>	<b>612.3</b>	<b>583.9</b>	<b>557.7</b>	<b>518.6</b>	<b>482.2</b>	<b>454.6</b>
110°	<b>675.0</b>	<b>605.3</b>	<b>546.4</b>	<b>513.4</b>	<b>488.1</b>	<b>461.6</b>	<b>407.7</b>
120°	<b>692.3</b>	<b>615.1</b>	<b>556.5</b>	<b>497.8</b>	<b>460.9</b>	<b>433.9</b>	<b>367.1</b>
135°	<b>762.1</b>	<b>646.8</b>	<b>590.5</b>	<b>536.4</b>	<b>481.7</b>	<b>425.0</b>	<b>302.3</b>
150°	<b>892.1</b>	<b>718.4</b>	<b>643.1</b>	<b>600.5</b>	<b>558.9</b>	<b>502.8</b>	<b>354.5</b>
Run VMG	<b>1030.1</b>	<b>829.5</b>	<b>741.7</b>	<b>680.3</b>	<b>628.9</b>	<b>580.6</b>	<b>409.4</b>
<b>Selected Courses</b>							
Windward / Leeward	<b>1126.9</b>	<b>925.8</b>	<b>829.0</b>	<b>777.4</b>	<b>746.0</b>	<b>720.2</b>	<b>632.6</b>
Circular Random	<b>967.6</b>	<b>791.8</b>	<b>698.2</b>	<b>641.9</b>	<b>603.0</b>	<b>571.4</b>	<b>515.2</b>
Ocean for PCS	<b>1180.5</b>	<b>924.7</b>	<b>780.4</b>	<b>689.0</b>	<b>623.7</b>	<b>570.7</b>	<b>482.1</b>
Non Spinnaker	<b>1053.3</b>	<b>851.0</b>	<b>741.9</b>	<b>677.4</b>	<b>635.7</b>	<b>605.7</b>	<b>560.4</b>
<b>Velocity Prediction in Knots for True Wind Speeds</b>							
Wind Velocity	6 kt	8 kt	10 kt	12 kt	14 kt	16 kt	20 kt
Beat Angles	<b>42.9°</b>	<b>42.1°</b>	<b>40.1°</b>	<b>38.8°</b>	<b>38.4°</b>	<b>38.5°</b>	<b>39.1°</b>
Beat VMG	<b>2.94</b>	<b>3.52</b>	<b>3.93</b>	<b>4.12</b>	<b>4.17</b>	<b>4.19</b>	<b>4.21</b>
52°	<b>4.50</b>	<b>5.27</b>	<b>5.63</b>	<b>5.82</b>	<b>5.90</b>	<b>5.95</b>	<b>6.07</b>
60°	<b>4.78</b>	<b>5.46</b>	<b>5.80</b>	<b>6.07</b>	<b>6.20</b>	<b>6.27</b>	<b>6.44</b>
75°	<b>5.02</b>	<b>5.61</b>	<b>5.97</b>	<b>6.39</b>	<b>6.73</b>	<b>6.89</b>	<b>7.03</b>
90°	<b>5.29</b>	<b>5.88</b>	<b>6.17</b>	<b>6.45</b>	<b>6.94</b>	<b>7.47</b>	<b>7.92</b>
110°	<b>5.33</b>	<b>5.95</b>	<b>6.59</b>	<b>7.01</b>	<b>7.38</b>	<b>7.80</b>	<b>8.83</b>
120°	<b>5.20</b>	<b>5.85</b>	<b>6.47</b>	<b>7.23</b>	<b>7.81</b>	<b>8.30</b>	<b>9.81</b>
135°	<b>4.72</b>	<b>5.57</b>	<b>6.10</b>	<b>6.71</b>	<b>7.47</b>	<b>8.47</b>	<b>11.91</b>
150°	<b>4.04</b>	<b>5.01</b>	<b>5.60</b>	<b>5.99</b>	<b>6.44</b>	<b>7.16</b>	<b>10.15</b>
Run VMG	<b>3.49</b>	<b>4.34</b>	<b>4.85</b>	<b>5.29</b>	<b>5.72</b>	<b>6.20</b>	<b>8.79</b>
Gybe Angles	<b>146.2°</b>	<b>149.0°</b>	<b>150.9°</b>	<b>175.6°</b>	<b>177.7°</b>	<b>142.6°</b>	<b>139.4°</b>
<b>Certificate</b> Number <b>SB20</b> ORC Ref <b>N/A</b> Issued On <b>19/01/2017</b> VPP Ver. <b>2017 1.00</b> <b>Invalid for Racing</b>							
<b>Crew Weight</b> Declared <b>270kg</b> Default* <b>312kg</b> Non Manual Pwr <b>No</b>							
<b>Special Scoring</b> ToD ToT Double H.GPH <b>730.0 0.8219</b> Double H.OSN <b>716.1 0.8378</b> Non Spin GPH <b>764.2 0.7851</b> Non Spin OSN <b>741.0 0.8097</b>							
<b>Sails Limitations</b> Headsails <b>5</b> Spinnakers <b>3</b>							
<b>Class Division Length</b> CDL = <b>5.872</b>							
<b>Storm Sails Areas</b> Heavy Weather Jib <b>7.65</b> Storm Jib (JL=4.89) <b>2.84</b> Storm Triesail <b>4.45</b>							
<b>Owner</b> I certify that I understand my responsibilities under ORC Rules and Regulations Signature							

<b>BOAT</b> Name <b>SB 20</b> Sail Nr <b>SB20-</b> File <b>SB20</b> Data in <b>meters/kilograms</b>		<b>INCLINING TEST AND FREEBOARDS</b> Inclining Test Current Inclining Flotation date <b>18/05/2016</b> SG <b>1.0230</b> FFM <b>0.711</b> FF <b>0.711</b> SFFP <b>0.150</b> FAM <b>0.464</b> FA <b>0.464</b> SAFP <b>6.058</b> W1 <b>17.9</b> PD1 <b>568.3</b> WD <b>6.160</b> W2 <b>17.9</b> PD2 <b>566.3</b> GSA <b>1.0</b> W3 <b>17.9</b> PD3 <b>569.1</b> RSA <b>1.0</b> W4 <b>17.9</b> PD4 <b>565.9</b> PLM <b>9000.0</b> LCF from stem on CL / on sheer <b>3.473 / 3.587</b> Maximum beam station from stem <b>3.900</b> RM Measured <b>15.3kg-m</b> RM Default <b>15.9kg-m</b> Limit of positive stability / Stab.Index <b>106.1° / 97.0</b> Freeboard at mast at 2.230 <b>0.597</b>		 World Leader in Rating Technology <b>2017</b> <b>IMS Measurement Certificate</b> Certificate Number <b>SB20</b> ORC Ref <b>N/A</b> Issued On <b>18/01/2017</b> VPP Ver. <b>2017 1.00</b> <b>Invalid for Racing</b>																				
<b>RIG</b> Forestay Tension <b>Aft</b> Spreaders <b>2</b> Inner Stay <b>None Fitted</b> Runners <b>0</b> Carbon Mast <b>No</b> Jumper Struts <b>None</b> Taper Hollows <b>No</b> Jib Furler <b>No</b> Fiber Rigging <b>No</b> Main Furler <b>No</b> Lenticular Rigging <b>No</b> Without Backstay <b>No</b> Articulated Bowsprit <b>No</b>		<b>PROPELLER</b> Type <b>No Propeller</b>																						
<table border="1"> <tr> <td>P <b>7.950</b></td> <td>E <b>3.200</b></td> <td>MDT1 <b>0.070</b></td> <td>MW <b>0.090</b></td> </tr> <tr> <td>IG <b>7.340</b></td> <td>J <b>2.180</b></td> <td>MDL1 <b>0.100</b></td> <td>GO <b>0.145</b></td> </tr> <tr> <td>ISP <b>8.940</b></td> <td>SFJ <b>0.050</b></td> <td>MDT2 <b>0.070</b></td> <td>BD <b>0.140</b></td> </tr> <tr> <td>BAS <b>0.940</b></td> <td>SPL <b>0.000</b></td> <td>MDL2 <b>0.082</b></td> <td>MWT <b>31.10</b></td> </tr> <tr> <td>FSP <b>4.050</b></td> <td>TPS <b>4.050</b></td> <td>TL <b>1.400</b></td> <td>MCG <b>3.290</b></td> </tr> </table>			P <b>7.950</b>		E <b>3.200</b>	MDT1 <b>0.070</b>	MW <b>0.090</b>	IG <b>7.340</b>	J <b>2.180</b>	MDL1 <b>0.100</b>	GO <b>0.145</b>	ISP <b>8.940</b>	SFJ <b>0.050</b>	MDT2 <b>0.070</b>	BD <b>0.140</b>	BAS <b>0.940</b>	SPL <b>0.000</b>	MDL2 <b>0.082</b>	MWT <b>31.10</b>	FSP <b>4.050</b>	TPS <b>4.050</b>	TL <b>1.400</b>	MCG <b>3.290</b>	<b>MOVEABLE BALLAST</b> N/A
P <b>7.950</b>	E <b>3.200</b>		MDT1 <b>0.070</b>		MW <b>0.090</b>																			
IG <b>7.340</b>	J <b>2.180</b>	MDL1 <b>0.100</b>	GO <b>0.145</b>																					
ISP <b>8.940</b>	SFJ <b>0.050</b>	MDT2 <b>0.070</b>	BD <b>0.140</b>																					
BAS <b>0.940</b>	SPL <b>0.000</b>	MDL2 <b>0.082</b>	MWT <b>31.10</b>																					
FSP <b>4.050</b>	TPS <b>4.050</b>	TL <b>1.400</b>	MCG <b>3.290</b>																					
<b>MIZZEN RIG AND SAILS</b> N/A		<b>CENTERBOARD</b> N/A																						
<b>COMMENTS</b> 																								

SAILS (Maximum Areas)									
Mainsail	MHB	MUW	MTW	MHW	MQW	Area	Area (r)	Formula	
	0.230	0.81	1.43	2.32	2.90	16.61	17.28	P/B · (E + 2 · MQW + 2 · MHW + 1.5 · MTW + MUW + 0.5 · MHB)	
Symmetric									
Not Available									
Asymmetric	SLU	SLE	SL	SHW	SFL	Area			
	9.55	9.55	9.55	5.64	5.40	44.50	AS · (SFL + 4 · SHW) / 6		

HEADSAILS									
Area = 0.1125 · HLU · (1.445 · HLP + 2 · HQW + 2 · HHW + 1.5 · HTW + HUW + 0.5 · HHB)									
HHB	HUW	HTW	HHW	HQW	HLP	HLU	Area	Btn	Fly
0.04	0.42	0.72	1.32	1.84	2.22	7.13	8.86	N	N
Meas.Date Material Comment									
Unknown									





## Anexo II - Tabela de velocidades do veleiro “*Alba*”

	A	B	C	D	E	F	G	H	I	J
1	<b>TWA\TWS</b>	<b>0</b>	<b>6</b>	<b>8</b>	<b>10</b>	<b>12</b>	<b>14</b>	<b>16</b>	<b>20</b>	<b>25</b>
2	<b>0</b>	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
3	<b>45</b>	0,0	1,0	2,0	3,0	3,2	3,5	3,6	4,0	4,2
4	<b>60</b>	0,0	1,2	3,0	4,0	4,5	5,0	5,3	5,5	5,8
5	<b>75</b>	0,0	1,6	3,2	4,7	5,4	6,0	6,3	6,5	6,8
6	<b>90</b>	0,0	2,0	3,2	5,0	5,5	6,0	6,5	7,0	7,2
7	<b>105</b>	0,0	1,8	3,0	4,8	5,4	5,6	6,3	6,8	7,2
8	<b>120</b>	0,0	1,6	2,8	4,0	5,2	5,4	6,0	6,5	7,0
9	<b>135</b>	0,0	1,4	2,6	3,8	4,8	5,0	5,8	6,2	6,7
10	<b>150</b>	0,0	1,3	2,4	3,6	4,4	4,8	5,5	5,8	6,4
11	<b>165</b>	0,0	1,2	2,2	3,2	4,2	4,7	5,3	5,6	6,2
12	<b>180</b>	0,0	1,0	2,2	3,0	4,0	4,5	5,0	5,5	6,0